

AD-A285 281

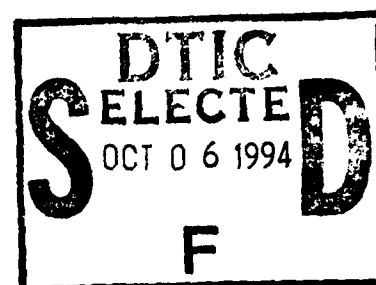


NAVAL HEALTH RESEARCH CENTER

HAZARDOUS MATERIAL LIFE-CYCLE COST MODEL

TECHNICAL MANUAL

VERSION 1.2



H. L. Ly

G. Pang

DTIC QUALITY INSPECTED 2

Technical Document 93-3C

17318
94-31770

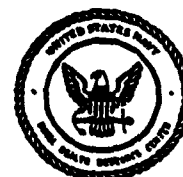
5 20

Approved for public release: distribution unlimited.



NAVAL HEALTH RESEARCH CENTER
P. O. BOX 85122
SAN DIEGO, CALIFORNIA 92186 - 5122

NAVAL MEDICAL RESEARCH AND DEVELOPMENT COMMAND
BETHESDA, MARYLAND



Hazardous Material Life-Cycle Cost Model

Technical Manual

Version 1.2

Prepared by:

Hoa Le Ly
Gerald Pang

Naval Health Research Center
Medical Information Systems and
Operations Research Department
P.O. Box 85122
San Diego, CA 92186-5122

Accession For	
NTIS CRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Technical Document 93-3C was supported by the Naval Medical Research and Development Command, Bethesda, MD, Department of the Navy, under a NAVSUP Reimbursable Work Unit. The views expressed in this article are those of the authors and do not reflect the official policy or position of the Department of the Navy, Department of Defense, nor the U.S. Government.

SUMMARY

This technical manual contains the information on the program source code, data elements, and file structures needed to maintain the Hazardous Material Life-Cycle Cost Model. This documentation was created using the FoxDoc Version 2.5a program.

TABLE OF CONTENTS

Introduction	1
Section I. System Summary	1
Section II. Menu Summary	1
Section III. Screen Summary	3
Section IV. Data Dictionary	12
A. Database Structure Summary	13
B. Database Field Summary	20
Section V. Tree Diagram	22
Section VI. Procedure and Function Summary	30
Section VII. Program Source Code	44

Introduction. This document was created using the FoxDoc Version 2.5a program to generate the technical documentation for the Hazardous Material Life-Cycle Cost Model (HMLCCM) system. The documentation is separated into seven sections: (1) System Summary, (2) Menu Summary, (3) Screen Summary, (4) Data Dictionary, (5) Tree Diagram, (6) Procedure Summary, and (7) Source Code Program Listings.

Section I. System Summary. See Section V for the tree diagram of the programs, procedures, functions, and format files.

```

This system has:
  11971 lines of code
    1 program file
    40 procedure files
  155 procedures and functions
    63 databases
    11 structural index files
    1 menu file
    11 screen files
    1 other file
  
```

Section II. Menu Summary. The following description lists the menu options used to drive the system. HMENU.MNX is the only menu template created.

<u>OPTION NAME</u>	<u>KEYS\FUNCTION CALLED</u>
System	ALT+S _MSM_SYSTM
Help...	F1 _MST_HELP
-----	_MST_SP100
Calculator	_MST_CALC
-----	(Submenu)
HMLCCM	(Submenu HMLCCM)
Cost Analysis	(Submenu COSTANALYS)
Build Hazmat Scenario	do hmssc
OSHA Z-Table	do hmztab
Reference Material	do hmmsds
System Maintenance	(Submenu SYSTEMMAIN)
Back-Up (floppy)	(Procedure)
Up-Load Data	(Procedure)
Set Parameters	(Submenu SETPARAMET)
Materials	CTRL+M do hmat.spr
Life Cycle Phase	CTRL+L do hmlc.spr
Process	CTRL+W do hmwp.spr
EXposure Type	CTRL+X do hmet.spr
Cost Factors	CTRL+F do hmcfe.spr
Cost Factor Elements	CTRL+E do hmcfe.spr
Cost Factor Element Items	CTRL+I do hmcfe.spr
Build Hazmat Table	CTRL+B do hmtab.spr
File	ALT+F _MSM_FILE
Printer Setup...	_MFI_SETUP
Print...	_MFI_PRINT
-----	_MFI_SP300
Quit	do _quit in hminit

<u>OPTION NAME</u>	<u>KEYS\FUNCTION CALLED</u>
Edit	ALT+E _MSM_EDIT
Undo	CTRL+U _MED_UNDO
Redo	CTRL+R _MED_REDO
-----	_MED_SP100
Cut	CTRL+X _MED_CUT
Copy	CTRL+C _MED_COPY
Paste	CTRL+V _MED_PASTE
Clear	_MED_CLEAR
-----	_MED_SP200
Select All	CTRL+A _MED_SLCTA
-----	_MED_SP300
Goto Line...	_MED_GOTO
Find...	CTRL+F _MED_FIND
Find Again	CTRL+G _MED_FINDA
Replace And Find Again	CTRL+E _MED_REPL
Replace All	_MED_REPLA
-----	_MED_SP400
Preferences...	_MED_PREF
Database	ALT+D _MSM_DATA
Browse	_MDA_BROW
-----	_MDA_SP100
Sort...	_MDA_SORT
Total...	_MDA_TOTAL
-----	_MDA_SP200
Average...	_MDA_AVG
Count...	_MDA_COUNT
Sum...	_MDA_SUM
Calculate...	_MDA_CALC
Report...	_MDA_REPRT
Record	ALT+R _MSM_REC RD
Goto...	_MRC_GOTO
Locate...	_MRC_LOCAT
Continue	CTRL+K _MRC_CONT
Seek...	_MRC_SEEK
-----	_MRC_SP200
Replace...	_MRC_REPL
Delete...	_MRC_DELET
Recall...	_MRC_RECAL
Program	ALT+P _MSM_PROG
Cancel	_MPR_CANCL
Resume	CTRL+M _MPR_RESUM
Window	ALT+W _MSM_WINDO
Hide	_MWI_HIDE
Hide All	_MWI_HIDEA
Show All	_MWI_SHOWA
Clear	_MWI_CLEAR
-----	_MWI_SP100
Move	CTRL+F7 _MWI_MOVE
Size	CTRL+F8 _MWI_SIZE
Zoom ^X	CTRL+F10 _MWI_ZOOM
Zoom ^Y	CTRL+F9 _MWI_MIN
Cycle	CTRL+F1 _MWI_ROTAT
Color...	_MWI_COLOR
-----	_MWI_SP200
Command	CTRL+F2 _MWI_CMD
Debug	_MWI_DEBUG
Trace	_MWI_TRACE
View	_MWI_VIEW

Section III. Screen Summary. Ten (10) screen files were created as input templates for the HMLCCM system: HMAT.SCX, HMLC.SCX, HMWP.SCX, HMET.SCX, HMC.F.SCX, HMC.FE.SCX, HMC.FEI.SCX, HMTAB.SCX, HMCOMP.SCX, and W_PRINT.SCX.

HMAT.SCX

Last updated: 09/15/93 at 13:34

HAZARDOUS MATERIALS

```

0  ID 1: hm   Materials Name:  2: hmatname.....
1  Manufactory: 3: mfg.....
2  NIIN #: 4: niin...   Common Name:  5: hmcom.....
3
46:msds.....
5.....
6.....
7.....
8.....
9.....
10.....
11.....
12.....
13.....
14.....
15.....
16.....
17
18      < Add  > < Next > <Previous> < Exit > < Save > <Cancel>

```

Window name: Hmat
Coordinates: FROM 0,0 TO 0,79
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: M.hmatid	Field	
2: M.hmatname	Field	"@!"
3: m.mfg	Field	
4: m.niin	Field	
5: m.hmcom	Field	
6: m.msds	Field	
7: m.Action	Push button	"@*HN \<Add;\<Next;\<Previous; \?E\<xit"
8: m.Save	Push button	"@*HN \<Save;\<Cancel"

HMLC.SCX

Last updated: 09/15/93 at 13:34

```
0          HM LIFE CYCLE PHASES
1
2          PHASE ID: 2: hmlcid.
3
4          PHASE NAME: 3: hmlc.....
5
6
7
8          < Save > <Cancel>
9
```

< Add >
< Edit >
« Next »
<Previous>
< Exit >

Window name: Hmlc
Coordinates: FROM 0,0 TO 0,51
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \<Add;\<Edit;\\!\<Next;\<Previous;\<E\<xit"
2: m.hmlcid	Field	
3: m.hmlc	Field	"@!"
4: m.Save	Push button	"@*HN \<Save;\<Cancel"

HMWP.SCX

Last updated: 09/15/93 at 13:34

```
0          HM PROCESSES
1
2          ID NUM: 2: hmwpid.
3
4          PROCESS: 3: hmw.....
5          .....
6          .....
7          .....
8          .....
9          .....
10
11          < Save > <Cancel>
12
```

< Add >
< Edit >
« Next »
<Previous>
< Exit >

Window name: Hmwp
Coordinates: FROM 0,0 TO 0,63
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \<Add;\<Edit;\\!\<Next;\<Previous;\<E\<xit"
2: m.hmwpid	Field	
3: m.hmwp	Field	"@!"
4: m.Save	Push button	"@*HN \<Save;\<Cancel"

HMET.SCX

Last updated: 09/15/93 at 13:34

```
0          HM EXPOSURE TYPES
1
2  ID NUM: 3: hmetid.
3
4  TYPE:.. 4: hmet.....
5
6
7
8      < Save > <Cancel>
9
```

< Add >
< Edit >
« Next »
<Previous>
< Exit >

Window name: Hmet
Coordinates: FROM 0,0 TO 0,51
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \<Add;\<Edit;\\!\<Next;\<Previous;\?E\<xit"
2: m.Save	Push button	"@*HN \<Save;\<Cancel"
3: m.hmetid	Field	
4: m.hmet	Field	"@!"

HMCF.SCX

Last updated: 09/15/93 at 13:34

```
0          HM COST FACTORS
1
2  ID NUM: 2: hmcfid.
3
4  COST FACTOR: 3: hmcfc.....
5
6
7
8      < Save > <Cancel>
9
```

< Add >
< Edit >
« Next »
<Previous>
< Exit >

Window name: Hmcf
Coordinates: FROM 0,0 TO 0,52
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \<Add;\<Edit;\\!\<Next;\<Previous;\?E\<xit"
2: m.hmcfid	Field	
3: m.hmcfc	Field	"@!"
4: m.Save	Push button	"@*HN \<Save;\<Cancel"

```

0          HM COST FACTOR ELEMENTS
1
2          < Add  >
3
4          < Edit  >
5          FACTOR:  2: hm  3: answr..... < Next  >
6
7          ELEMENT: 4: hm  5: hmcfe..... <Previous>
8
9          < Save > <Cancel>
10
11
12
13          < Top   >
14
15          < Bottom >
16
17          < Exit  >

```

Window name: Hmcfe

Coordinates: FROM 0,0 TO 0,66

Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \<Add;\<Edit;\<Next;\<Previous; \!\<Top;\<Bottom;\?E\<xit"
2: m.hmcfid	Field	
3: m.answr	Field	
4: m.hmcfeid	Field	
5: m.hmcfe	Field	"@!"
6: m.Save	Push button	"@*HN \<Save;\<Cancel"

```

0          COST FACTOR ELEMENT ITEMS          < Add  >
1
2          FACTOR: 2: hm  3: answr.....          < Edit  >
3
4          ELEMENT: 4: hm  5: hmcfe.....          < Next  >
5
6          ITEM: 6: hm  7: hmcfei.....          <Previous>
7
8          COST:  8: hmcfeic          « Top  »
9
10          < Save > <Cancel>          < Bottom >
11
12          < Browse >
13
14          < Exit  >

```

Window name: Hmcfei

Coordinates: FROM 0,0 TO 0,68

Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \<Add;\<Edit;\<Next;\<Previous; \!\<Top;\<Bottom;B\<rowse;\?E\<xit"
2: m.hmcfid	Field	
3: m.answr	Field	
4: m.hmcfeid	Field	
5: m.hmcfe	Field	"@!"
6: M.hmcfeiid	Field	
7: M.hmcfei	Field	"@!"
8: M.hmcfeicost	Field	"@\$"
9: m.Save	Push button	"@*HN \<Save;\<Cancel"

HMTAB.SCX

Last updated: 09/15/93 at 13:34

0	#2: t	HAZARDOUS MATERIALS TABLE	
1	Material: 3: h 4: hmname.....		< Add >
2 Per: 5: hmunit.....		
3		< Edit >
4	Phase: 6: h 7: hmlc.....		
5	Process: 8: h 9: hmwpid.....		< Next >
6		
7	Exposure Type: 10: 11: hmet.....		< Previous >
8	Probability of exposure: 12: hmetp		
9			< Top >
10	Factor: 13: 14: hmcfe.....		
11	Element: 15: 16: hmcfe.....		< Bottom >
12	Item: 17: 18: hmcfeid.....		
13			< Exit >
14	Cost: 19: hmcfe	Probability 20: p	
15			
16	By Person	() YES(■) NO	< Save >
17	By Day	() YES(■) NO	
18	By Quantity	() YES(■) NO	< Cancel >
19			

Window name: Hmtab

Coordinates: FROM 0,0 TO 0,78

Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \<Add;\<Edit;\.\<Next;\<Previous;\<Top;\<Bottom;\?E\<xit"
2: m.tabid	Field	"@Z"
3: m.hmatid	Field	"@!"
4: m.hmname	Field	"@!"
5: m.hmunit	Popup	"@^"
6: m.hmlcid	Field	"@Z"
7: m.hmlc	Field	"@!"
8: m.hmwpid	Field	"@Z"
9: m.hmwpid	Field	"@!"
10: m.hmetid	Field	"@Z"
11: m.hmet	Field	"@!"
12: m.hmetprob	Field	"@Z"
13: m.hmcfeid	Field	"@Z"
14: m.hmcfe	Field	"@Z"
15: m.hmcfeid	Field	"@Z"
16: m.hmcfe	Field	"@!"
17: m.hmcfeid	Field	"@Z"
18: m.hmcfe	Field	"@!"
19: m.hmcfecost	Field	"@\$\$\$,\$\$\$\$.99"
20: m.prob	Field	"9.99"
21: m.perp	Radio button	"@*RHN YES;NO"
22: m.perd	Radio button	"@*RHN YES;NO"
23: m.perq	Radio button	"@*RHN YES;NO"
24: m.Save	Push button	"@*VN \<Save;\<Cancel"

COMPUTE COST VALUE

0	Estimated		Estimated
1	Cost	Scenario:1: 2: chmscname...	Variance
2			
3	for Product:	3: hmat.....	
4			
5			
6	[] . for Step:	4: step.....	[] .
7			
8			
9	[] . for Factor:	7: fact.....	[] .
10			
11			
12	[] . for Factor:	10: wfact..... at step: 11: wstep....	[] .
13			
14			
15	[] . for Phase:	14: phase.....	[] .
16			
17	# of Iterations	Total of Scenario cost: () Yes (■) No	
18	18: sample..		
19		< Ok > <Browse> <Cancel>	

Window name: W_hmcost
 Coordinates: FROM 0,0 TO 0,75
 Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.chmscid	Field	"@Z"
2: m.chmscname	Field	"@!T"
3: m.hmat	Popup	"@^ "
4: m.step	Popup	"@^ "
5: m.cstep	Check box	"@*C "
6: m.bstep	Check box	"@*C "
7: m.fact	Popup	"@^ "
8: m.cfact	Check box	"@*C "
9: m.bfact	Check box	"@*C "
10: m.wfact	Popup	"@^ "
11: m.wstep	Popup	"@^ "
12: m.cwstep	Check box	"@*C "
13: m.bwstep	Check box	"@*C "
14: m.phase	Popup	"@^ "
15: m.cphase	Check box	"@*C "
16: m.bphase	Check box	"@*C "
17: m.scost	Radio button	"@*RHN Yes ;No"
18: m.sample	Field	"@Z"
19: m.action	Push button	"@*HN \<Ok;\<Browse;\<Cancel"

W_PRINT.SCX

Last updated: 09/15/93 at 13:34

Print Option

```
0
1  Print to:.      (■) File
2                  ( ) Printer
3
4
5
6  Filename: 1: mprintfile.....
7
8
9
10      < OK >   <Cancel>
```

Window name: W_prn
Coordinates: FROM 0,0 TO 0,42
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mprintfile	Field	
2: mbuttons	Push button	"@*HT OK;\?Cancel"
3: mchoice	Radio button	"@*RVN File;Printer"

Section IV. Data Dictionary. There are eighteen database files in the HMLCCM system:

HMAT.DBF	-- Hazardous Materials File
HMCOM.DBF	-- Hazardous Material Common Name Table
HMLC.DBF	-- Hazardous Material Life Cycle Table
HMWP.DBF	-- Hazardous Material Work Process Table
HMET.DBF	-- Hazardous Material Exposure Type Table
HMCF.DBF	-- Hazardous Material Cost Factor Table
HMCFE.DBF	-- Hazardous Material Cost Factor Element Table
HMCFEI.DBF	-- Hazardous Material Cost Factor Element Item Table
HMUNIT.DBF	-- Hazardous Material Unit Table
HMTAB.DBF	-- Hazardous Material Scenario Table
HMSTEP.DBF	-- Hazardous Material Step Table
HMCOMP.DBF	-- Hazardous Material Cost Computation File
HMSCEN.DBF	-- Hazardous Material Scenario File
BOOT.DBF	-- Temporary file used for the bootstrap computations
HMTEMP.DBF	-- Temporary file used to store data from tables
TEMP.DBF	-- Temporary file used to store data from tables
TEST.DBF	-- Temporary file used to store data from tables
TLV	-- Threshold Limit Values for OSHA Z-Table

A. Database Structure Summary.

Structure for database : **HMAT.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	HMATID	Numeric	4		1	4
2	HMATNAME	Character	45		5	49
3	MFG	Character	25		50	74
4	NIIN	Character	9		75	83
5	MILSPEC	Character	11		84	94
6	HMCOST	Numeric	10	2	95	104
7	HMUNIT	Character	10		105	114
8	HMDESCRIPT	Memo	10		115	124
9	CHEM_PHY	Memo	10		125	134
10	HEALTH_HAZ	Memo	10		135	144
11	MED_SURV	Memo	10		145	154
12	OCC_EXP	Memo	10		155	164
13	PPE_TREAT	Memo	10		165	174
14	SPEC_TESTS	Memo	10		175	184
15	TREATMENT	Memo	10		185	194
16	SYN_TRADE	Memo	10		195	204
17	COM_USES	Memo	10		205	214
18	EXP_LIMITS	Memo	10		215	224
19	MSDS	Memo	10		225	234
** Total **			235			

This database is associated with the memo file: HMAT.FPT

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\HMAT.CDX

Used by: HMAT.SPR
: HMLU.PRG
: GET_HMCOMID() (function in C:\HAZMAT\GHM\WORK\HMTAB.SPR)
: GET_HMNAME (procedure in C:\HAZMAT\GHM\WORK\HMLU.PRG)

Structure for database : **HMCOM.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	HMCOMID	Numeric	4		1	4
2	COMMON	Character	30		5	34
3	NIIN	Character	9		35	43
** Total **			44			

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\HMCOM.CDX

Used by: COMNAME() (function in C:\HAZMAT\GHM\WORK\HMAT.SPR)
: SAVECOM (procedure in C:\HAZMAT\GHM\WORK\HMAT.SPR)
: GET_HMCOMID() (function in C:\HAZMAT\GHM\WORK\HMTAB.SPR)
: COMPUTSTEP() (function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)
: COMPUTMAT() (function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)
: COMPUTFACT() (function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)
: COMPUTWFACT() (function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)

: COMPUTPHASE()	(function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)
: BSTEP()	(function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)
: BHMAT()	(function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)
: BFACT()	(function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)
: BWSTEP()	(function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)
: BPHASE()	(function in C:\HAZMAT\GHM\WORK\HMCOMP.SPR)

Structure for database : **HMLC.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	HMLCID	Numeric	2		1	2
2	HMLC	Character	30		3	32
** Total **			33			

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\HMLC.CDX

Used by: HMLC.SPR

Structure for database : **HMWP.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	HMWPID	Numeric	2		1	2
2	HMWP	Character	80		3	82
3	HMPLATFORM	Numeric	2		83	84
** Total **			85			

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\HMWP.CDX

Used by: HMWP.SPR

Structure for database : **HMET.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	HMETID	Numeric	2		1	2
2	HMET	Character	30		3	32
** Total **			33			

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\HMET.CDX

Used by: HMET.SPR

Structure for database : **HMCF.DBF** Alias: CFTMP

Field	Field name	Type	Width	Dec	Start	End
1	HMCFID	Numeric	2		1	2
2	HMCF	Character	30		3	32
** Total **			33			

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\HMCF.CDX

Used by: HMCF.SPR

: REL()	(function in C:\HAZMAT\GHM\WORK\HMCFE.SPR)
: GET_HMCFE	(procedure in C:\HAZMAT\GHM\WORK\HMCFEI.SPR)
: GET_CFE	(procedure in C:\HAZMAT\GHM\WORK\HMCFEI.SPR)
: GET_CFEID	(procedure in C:\HAZMAT\GHM\WORK\HMCFEI.SPR)
: BROWSEITEM	(procedure in C:\HAZMAT\GHM\WORK\HMCFEI.SPR)
: GET_CFar1()	(function in C:\HAZMAT\GHM\WORK\HMTAB.SPR)
: GET_CF1()	(function in C:\HAZMAT\GHM\WORK\HMTAB.SPR)
: GET_EID()	(function in C:\HAZMAT\GHM\WORK\HMTAB.SPR)

Structure for database : **HMCFE.DBF** Alias: CFETMP

Field	Field name	Type	Width	Dec	Start	End
1	HMCFEID	Numeric	2		1	2
2	HMCFID	Numeric	2		3	4
3	HMCFE	Character	30		5	34
** Total **			35			

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\HMCFE.CDX

Used by: HMCFE.SPR

: REL()	(function in C:\HAZMAT\GHM\WORK\HMCFE.SPR)
: BROWSEITEM	(procedure in C:\HAZMAT\GHM\WORK\HMCFEI.SPR)
: GET_EI()	(function in C:\HAZMAT\GHM\WORK\HMTAB.SPR)
: GET_EIID()	(function in C:\HAZMAT\GHM\WORK\HMTAB.SPR)

Structure for database : **HMCFEI.DBF**

Alias: CFEITMP

Field	Field name	Type	Width	Dec	Start	End
1	HMCFEIID	Numeric	4		1	4
2	HMCFEID	Numeric	2		5	6
3	HMCFID	Numeric	2		7	8
4	HMCFEI	Character	50		9	58
5	HMCFEINO	Character	15		59	73
6	HMCFEICOST	Numeric	10	2	74	83
7	HMCFEIUNIT	Character	10		84	93
** Total **			94			

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\HMCFEI.CDX

Used by: HMCFEI.SPR
: REL() (function in C:\HAZMAT\GHM\WORK\HMCFE.SPR)
: BROWSEITEM (procedure in C:\HAZMAT\GHM\WORK\HMCFEI.SPR)

Structure for database : **HMUNIT.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	HMUNIT	Character	20		1	20
** Total **			21			

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\HMUNIT.CDX

Used by: HMTAB.SPR

Structure for database : **HMTAB.DBF**

Alias: HMTAB

Field	Field name	Type	Width	Dec	Start	End
1	TABID	Numeric	4		1	4
2	HMCOMID	Numeric	4		5	8
3	HMATID	Numeric	4		9	12
4	HMLCID	Numeric	2		13	14
5	HMWPID	Numeric	2		15	16
6	HMETID	Numeric	2		17	18
7	HMETPROB	Float	5		19	23
8	HMCFID	Numeric	2		24	25
9	HMCFEID	Numeric	2		26	27
10	HMCFEIID	Numeric	4		28	29
11	PERP	Numeric	1		30	30
12	PERD	Numeric	1		31	31
13	PERQ	Numeric	1		32	32
14	HMCFECOST	Numeric	5	2	33	37
15	HMUNIT	Character	10		38	47
16	WTAVERAGE	Float	5		48	52
17	PROB	Float	5		53	57
** Total **			58			

FoxDoc did not find any associated index files

FoxDoc did not find any associated multiple indexes

Used by: HMTAB.SPR

: CHANGE

(procedure in C:\HAZMAT\GHM\WORK\HMTAB.SPR)

Structure for database : **HMSTEP.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	HMSCID	Numeric	4		1	4
2	HMSTEP	Numeric	4		5	8
3	HMATID	Numeric	4		9	12
4	HMLCID	Numeric	2		13	14
5	HMWPID	Numeric	2		15	16
6	PERNUM	Numeric	3		17	19
7	DURNUM	Numeric	3		20	22
8	QTYNUM	Numeric	3		23	25
9	WSTNUM	Numeric	3		26	28
10	UNIT	Character	20		29	48
** Total **			49			

Used by: GET_HMARRAY

: HMSTEP.PRG

(procedure in HMCOMP.SPR)

Structure for database : **HMCOMP.DBF** Status: Temporary

Used by: HMCOMP.SPR
: COMPUTSTEP() (function in HMCOMP.SPR)
: COMPUTFACT() (function in HMCOMP.SPR)
: COMPUTWFACT() (function in HMCOMP.SPR)
: COMPUTPHASE() (function in HMCOMP.SPR)
: BSTEP() (function in HMCOMP.SPR)
: BFACT() (function in HMCOMP.SPR)
: BWSTEP() (function in HMCOMP.SPR)
: BPHASE() (function in HMCOMP.SPR)

Structure for database : **HMSCEN.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	HMSCID	Numeric	4		1	4
2	HMSCNAME	Character	40		5	44
** Total **			45			

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\HMSCEN.CDX

Used by: HMSCEN.PRG
: SCSAVE (procedure in C:\HAZMAT\GHM\WORK\HMSTEP.PRG)

Structure for database: **BOOT.DBF** Status: Temporary File

Used by: BSTEP() (function in HMCOMP.SPR)
: BFACT() (function in HMCOMP.SPR)
: BWSTEP() (function in HMCOMP.SPR)
: BPHASE() (function in HMCOMP.SPR)

Structure for database : **HMTEMP.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	HMSCID	Numeric	4		1	4
2	HMSTEP	Numeric	4		5	8
3	HMATID	Numeric	4		9	12
4	HMLCID	Numeric	2		13	14
5	HMWPID	Numeric	2		15	16
6	PERNUM	Numeric	3		17	19
7	DURNUM	Numeric	3		20	22
8	QTYNUM	Numeric	3		23	25
9	WSTNUM	Numeric	3		26	28
10	UNIT	Character	20		29	48
** Total **			49			

Used by: SCSAVE (procedure in HMSTEP.PRG)

Structure for database : **TEMP.DBF** Status: Temporary File

Used by: GET_TABLE (procedure in HMCOMP.SPR)

Structure for database : **TEST.DBF** Status: Temporary File

Used by: CALCULATE() (function in HMCOMP.SPR)

Structure for database : **TLV.DBF**

Field	Field name	Type	Width	Dec	Start	End
1	SUBSTANCE	Character	40		1	40
2	GENERIC	Character	35		41	75
3	CHEMICAL	Character	35		76	110
4	CAS_NO	Character	12		111	122
5	NAV_TWA	Character	6		123	128
6	NAV_STEL	Character	6		129	134
7	NAV_CEIL	Character	6		135	140
8	NAV_SOURCE	Character	5		141	145
9	TLV_TWA	Character	6		146	151
10	TLV_C	Character	6		152	157
11	SKIN	Logical	1		158	158
12	TLV_STEL	Character	6		159	164
13	APPENDIX	Character	2		165	166
14	PEL_TWA	Character	6		167	172
15	PEL_C	Character	6		173	178
16	PEL_STEL	Character	6		179	184
17	OSHA_SKIN	Logical	1		185	185
18	PTLV_TWA	Character	6		186	191
19	PTLV_STEL	Character	6		192	197
20	PTLV_C	Character	6		198	203
21	NOTIFY	Logical	1		204	204
22	BASIS	Character	35		205	239
23	REPRO	Character	10		240	249
24	S_BASIS	Character	10		250	259
** Total **			260			

FoxDoc did not find any associated index files

This database appears to be associated with multiple index file(s):
: C:\HAZMAT\GHM\WORK\TLV.CDX

Used by: HMZTAB.PRG

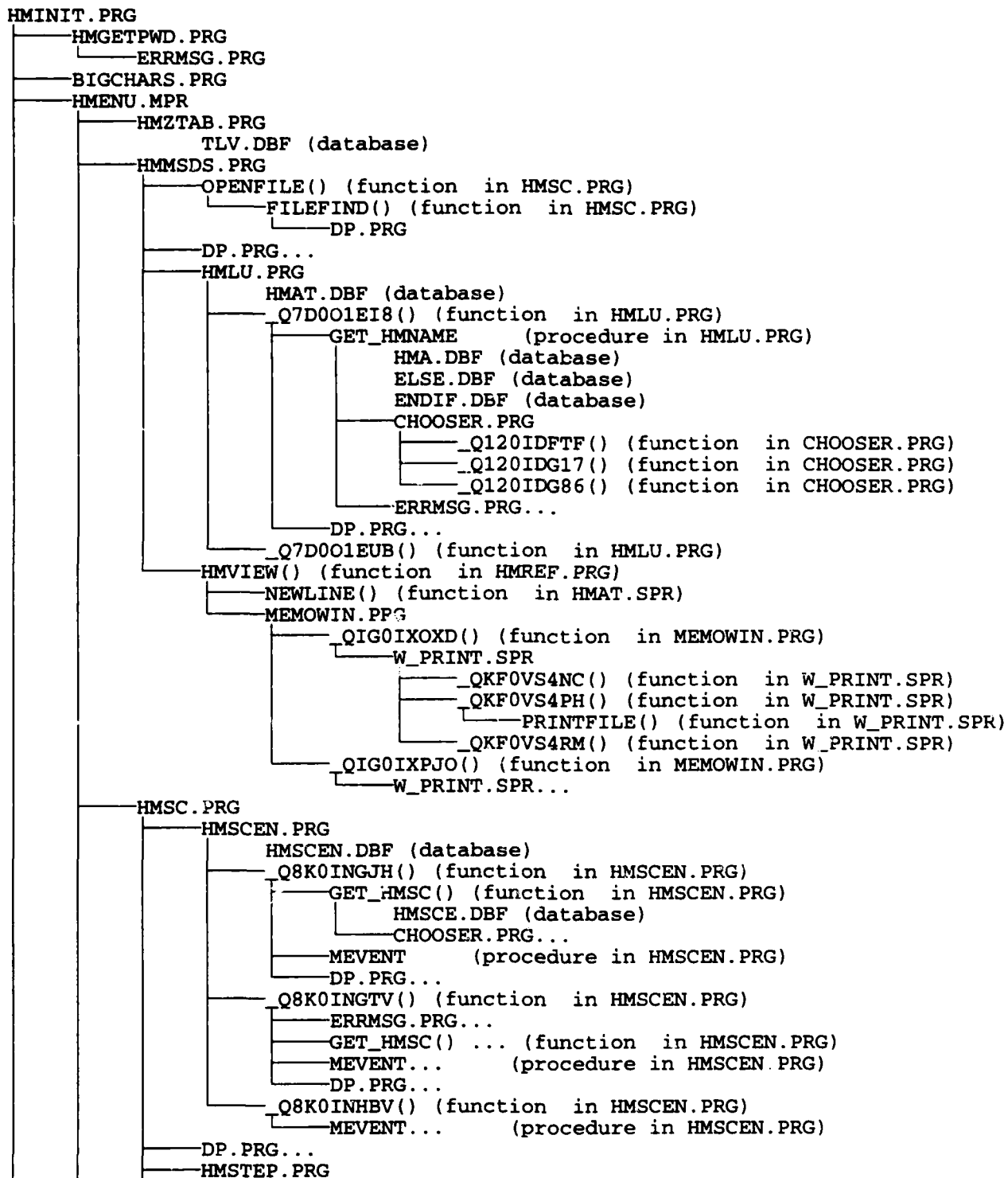
B. Database Field Summary

Field Name	Type	Len	Dec	Database
APPENDIX	C	2	0	TLV.DBF
BASIS	C	35	0	TLV.DBF
CAS_NO	C	12	0	TLV.DBF
CHEMICAL	C	35	0	TLV.DBF
CHEM_PHY	M	10	0	HMAT.DBF
COMMON	C	30	0	HMCOM.DBF
COM_USES	M	10	0	HMAT.DBF
EXP_LIMITS	M	10	0	HMAT.DBF

Field Name	Type	Len	Dec	Database
GENERIC	C	35	0	TLV.DBF
HEALTH_HAZ	M	10	0	HMAT.DBF
HMATID	N	4	0	HMAT.DBF
				HMTAB.DBF
HMATNAME	C	45	0	HMAT.DBF
HMCF	C	30	0	HMCF.DBF
HMCFE	C	30	0	HMCFE.DBF
HMCFECOST	N	5	2	HMTAB.DBF
HMCFEI	C	50	0	HMCFEI.DBF
HMCFEICOST	N	10	2	HMCFEI.DBF
HMCFEID	N	2	0	HMCFEI.DBF
				HMTAB.DBF
				HMCFE.DBF
HMCFEIID	N	4	0	HMCFEI.DBF
HMCFEIID	N	4	0	HMTAB.DBF
HMCFEINO	C	15	0	HMCFEI.DBF
HMCFEIUNIT	C	10	0	HMCFEI.DBF
HMCFID	N	2	0	HMCFEI.DBF
				HMTAB.DBF
				HMCFE.DBF
				HMCF.DBF
HMCOMID	N	4	0	HMTAB.DBF
				HMCOM.DBF
HMCOST	N	10	2	HMAT.DBF
HMDESCRIPT	M	10	0	HMAT.DBF
HMET	C	30	0	HMET.DBF
HMETID	N	2	0	HMTAB.DBF
				HMET.DBF
HMETPROB	F	5	3	HMTAB.DBF
HMLC	C	30	0	HMLC.DBF
HMLCID	N	2	0	HMLC.DBF
				HMTAB.DBF
HMPLATFORM	N	2	0	HMWP.DBF
HMSCID	N	4	0	HMSCEN.DBF
HMSCNAME	C	40	0	HMSCEN.DBF
HMUNIT	C	10	0	HMAT.DBF
				HMTAB.DBF
HMUNIT	C	20	0	HMUNIT.DBF
HMWP	C	80	0	HMWP.DBF
HMWPID	N	2	0	HMTAB.DBF
				HMWP.DBF
MED_SURV	M	10	0	HMAT.DBF
MFG	C	25	0	HMAT.DBF
MILSPEC	C	11	0	HMAT.DBF

MSDS	M	10	0	HMAT.DBF
NAV_CEIL	C	6	0	TLV.DBF
NAV_SOURCE	C	5	0	TLV.DBF
NAV_STEL	C	6	0	TLV.DBF
NAV_TWA	C	6	0	TLV.DBF
NIIN	C	9	0	HMCOM.DBF
				HMAT.DBF
NOTIFY	L	1	0	TLV.DBF
OCC_EXP	M	10	0	HMAT.DBF
OSHA_SKIN	L	1	0	TLV.DBF
PEL_C	C	6	0	TLV.DBF
PEL_STEL	C	6	0	TLV.DBF
PEL_TWA	C	6	0	TLV.DBF
Field Name	Type	Len	Dec	Database
PERD	N	1	0	HMTAB.DBF
PERP	N	1	0	HMTAB.DBF
PERQ	N	1	0	HMTAB.DBF
PPE_TREAT	M	10	0	HMAT.DBF
PROB	F	5	2	HMTAB.DBF
PTLV_C	C	6	0	TLV.DBF
PTLV_STEL	C	6	0	TLV.DBF
PTLV_TWA	C	6	0	TLV.DBF
REPRO	C	10	0	TLV.DBF
SKIN	L	1	0	TLV.DBF
SPEC_TESTS	M	10	0	HMAT.DBF
SUBSTANCE	C	40	0	TLV.DBF
SYN_TRADE	M	10	0	HMAT.DBF
S_BASIS	C	10	0	TLV.DBF
TABID	N	4	0	HMTAB.DBF
TLV_C	C	6	0	TLV.DBF
TLV_STEL	C	6	0	TLV.DBF
TLV_TWA	C	6	0	TLV.DBF
TREATMENT	M	10	0	HMAT.DBF
WTAVERAGE	F	5	2	HMTAB.DBF

Section V. Tree Diagram. The tree diagram lists each program that is called by the order in which it is used. Under each program there is a list of all functions called and the where the procedure/function is stored.



- HMSTE.DBF (database)
- ENDIF.DBF (database)
- OPENFILE() ... (function in HMSC.PRG)
- ERRMSG.PRG...
- CANCEL (procedure in HMSCP.SPR)
- ERRMSG.PRG...
- INITVAR (procedure in HMSTEP.PRG)
 - GET_HMATN() (function in HMSTEP.PRG)
 - HMCO.DBF (database)
 - GET_HMLCN() (function in HMTAB.SPR)
 - HML.DBF (database)
 - GET_HMWP() (function in HMTAB.SPR)
 - HMW.DBF (database)
- QIPOVASCN() (function in HMSTEP.PRG)
- YESNO.PRG
 - ERRMSG.PRG...
- INITVAR... (procedure in HMSTEP.PRG)
- ADDOPTION (procedure in HMSTEP.PRG)
- SCSAVE (procedure in HMSTEP.PRG)
- HMTEMP.DBF (database)
- HMSCEN.DBF (database)
- QIPOVASVS() (function in HMSTEP.PRG)
- QIPOVASYA() (function in HMSTEP.PRG)
 - GET_HMAT() (function in HMSTEP.PRG)
 - HMCO.DBF (database)
 - CHOOSE.PRG...
 - ERRMSG.PRG...
- DP.PRG...
- GET_HMUNIT (procedure in HMSTEP.PRG)
 - HMTA.DBF (database)
 - IF.DBF (database)
 - !EMPTY(X[1]).DBF (database)
 - ERRMSG.PRG...
- QIPOVAT7G() (function in HMSTEP.PRG)
- QIPOVAT9X() (function in HMSTEP.PRG)
 - GET_HMLC() (function in HMTAB.SPR)
 - HML.DBF (database)
 - CHOOSE.PRG...
 - ERRMSG.PRG...
- DP.PRG...
- GET_HMUNIT... (procedure in HMSTEP.PRG)
- QIPOVATIU() (function in HMSTEP.PRG)
- QIPOVATLC() (function in HMSTEP.PRG)
 - GET_HMWP() (function in HMTAB.SPR)
 - HMW.DBF (database)
 - CHOOSE.PRG...
 - ERRMSG.PRG...
- DP.PRG...
- GET_HMUNIT... (procedure in HMSTEP.PRG)
- QIPOVATWZ() (function in HMSTEP.PRG)
 - DATACHECK() (function in HMSTEP.PRG)
 - ERRMSG.PRG...
 - GET_HMUNIT... (procedure in HMSTEP.PRG)
- CHANGE (procedure in HMTAB.SPR)
- HMTAB.DBF (database)
- YESNO.PRG...
- INITVAR... (procedure in HMSTEP.PRG)
- ADDOPTION... (procedure in HMSTEP.PRG)
- QIPOVAUD2() (function in HMSTEP.PRG)

	DEL_HMSC() (function in HMSC.PRG)
	POPUPSHOW() (function in HMSC.PRG)
	OPENFILE() ... (function in HMSC.PRG)
	POPUPHIDE() (function in HMSC.PRG)
	YESNO.PRG...
	HMCOMP.SPR
	HMCOMP.DBF (database)
	OPEN (procedure in HMCOMP.SPR)
	OPENFILE() ... (function in HMSC.PRG)
	CANCEL... (procedure in HMCOMP.SPR)
	GET_HMARRAY (procedure in HMCOMP.SPR)
	HMSTE.DBF (database)
	GET_TABLE (procedure in HMCOMP.SPR)
	HMTA.DBF (database)
	TEMP.DBF (database)
	_QLE0L6B8T() (function in HMCOMP.SPR)
	_QLE0L6BIV() (function in HMCOMP.SPR)
	_QLE0L6BR7() (function in HMCOMP.SPR)
	_QLE0L6BYN() (function in HMCOMP.SPR)
	_QLE0L6C5N() (function in HMCOMP.SPR)
	_QLE0L6CB0() (function in HMCOMP.SPR)
	SETUPBOOT() (function in HMCOMP.SPR)
	BSELECT.PRG
	_Q8Q0N6SBM() (function in BSELECT.PRG)
	ERRMSG.PRG...
	BWFACT.PRG
	_Q8Q0NQKA6() (function in BWFACT.PRG)
	DP.PRG...
	POPUPSHOW() ... (function in HMSC.PRG)
	COMPUTE (procedure in HMCOMP.SPR)
	SUBCOMPUT (procedure in HMCOMP.SPR)
	COMPUTSTEP() (function in HMCOMP.SPR)
	HMCOM.DBF (database)
	M.SUM (database)
	=.DBF (database)
	CALCULATE(SN.DBF (database)
	SD.DBF (database)
	SQ).DBF (database)
	COMPUTFACT() (function in HMCOMP.SPR)
	HMCOM.DBF (database)
	M.SUM (database)
	=.DBF (database)
	CALCULATE(HMTEMP.PERNUM (database)
	HMTEMP.DURNUM (database)
	HMTEMP.QTYNUM) (database)
	COMPUTWFACT() (function in HMCOMP.SPR)
	HMCOM.DBF (database)
	M.RESUL (database)
	=.DBF (database)
	IIF(EMPTY(M.RESUL) (database)
	U).DBF (database)
	ALLTRIM(STR(HMTEMP.HMSTEP)) (database)
	ALLTRIM(STR(CALCULATE(HMTEMP.PERNUM
(database)	
	HMTEMP.DURNUM (database)
	HMTEMP.QTYNUM) (database)
	8.DBF (database)
	2)).DBF (database)
	COMPUTPHASE() (function in HMCOMP.SPR)



- COMNAME() (function in HMAT.SPR)
- HMCOM.DBF (database)
- QKF0VR394() (function in HMAT.SPR)
 - ERRMSG.PRG...
 - CHANGE... (procedure in HMAT.SPR)
- QKF0VR3EK() (function in HMAT.SPR)
 - CHANGE... (procedure in HMAT.SPR)
- QKF0VR3HM() (function in HMAT.SPR)
 - COMNAME() ... (function in HMAT.SPR)
 - CHANGE... (procedure in HMAT.SPR)
- QKF0VR3KQ() (function in HMAT.SPR)
- QKF0VR3ME() (function in HMAT.SPR)
- QKF0VR3ON() (function in HMAT.SPR)
 - CHANGE... (procedure in HMAT.SPR)
- QKF0VR3R9() (function in HMAT.SPR)
 - COMNAME() ... (function in HMAT.SPR)
- QKF0VR3XH() (function in HMAT.SPR)
 - ERRMSG.PRG...
 - CHANGE... (procedure in HMAT.SPR)
 - SAVECOM (procedure in HMAT.SPR)
 - HMCOM.DBF (database)
 - COMNAME() ... (function in HMAT.SPR)
- HMLC.SPR
 - HMLC.DBF (database)
 - QKF0VR6JX() (function in HMLC.SPR)
 - QKF0VR6TE() (function in HMLC.SPR)
 - ERRMSG.PRG...
 - QKF0VR6VW() (function in HMLC.SPR)
 - CHANGE... (procedure in HMAT.SPR)
- HMWP.SPR
 - HMWP.DBF (database)
 - QKF0VR92I() (function in HMWP.SPR)
 - QKF0VR99I() (function in HMWP.SPR)
 - ERRMSG.PRG...
 - QKF0VR9C5() (function in HMWP.SPR)
 - ERRMSG.PRG...
- HMET.SPR
 - HMET.DBF (database)
 - QKF0VRBIT() (function in HMET.SPR)
 - QKF0VRBPA() (function in HMET.SPR)
 - CHANGE... (procedure in HMAT.SPR)
 - QKF0VRBT3() (function in HMET.SPR)
 - ERRMSG.PRG...
- HMCF.SPR
 - HMCF.DBF (database)
 - QKF0VRDYQ() (function in HMCF.SPR)
 - QKF0VRE5A() (function in HMCF.SPR)
 - ERRMSG.PRG...
 - QKF0VRE7O() (function in HMCF.SPR)
 - ERRMSG.PRG...
- HMCFE.SPR
 - HMCFE.DBF (database)
 - GET_HMCF (procedure in HMCFE.SPR)
 - HMC.DBF (database)
 - M.HMCF=X[1] (database)
 - IF.DBF (database)
 - NOT.DBF (database)
 - EMPTY(X[1]).DBF (database)
 - QKF0VRGGZ() (function in HMCFE.SPR)

- GET_HMCF... (procedure in HMCFE.SPR)
- QKF0VRGRP() (function in HMCFE.SPR)
- QKF0VRGTP() (function in HMCFE.SPR)
- REL() (function in HMCFE.SPR)
 - HMCFEI.DBF (database)
 - HMCFE.DBF (database)
 - HMCF.DBF (database)
- GET_HMCF... (procedure in HMCFE.SPR)
- QKF0VRGX4() (function in HMCFE.SPR)
- QKF0VRGYG() (function in HMCFE.SPR)
- REL() ... (function in HMCFE.SPR)
- GET_HMCF... (procedure in HMCFE.SPR)
- QKF0VRH1G() (function in HMCFE.SPR)
- QKF0VRH34() (function in HMCFE.SPR)
- GET_CF() (function in HMCFE.SPR)
 - HMC.DBF (database)
 - CHOOSE.PRG...
 - ERRMSG.PRG...
- GET_CFID (procedure in HMCFE.SPR)
 - HMC.DBF (database)
 - X=X[1].DBF (database)
 - IF.DBF (database)
 - NOT.DBF (database)
 - EMPTY(X[1]).DBF (database)
- QKF0VRH6D() (function in HMCFE.SPR)
- GET_HMCF... (procedure in HMCFE.SPR)
- HMCFEI.SPR
 - HMCFEI.DBF (database)
 - GET_HMCF... (procedure in HMCFE.SPR)
 - GET_HMCFE (procedure in HMCFEI.SPR)
 - HMCF.DBF (database)
 - IF.DBF (database)
 - !EMPTY(X[1]).DBF (database)
 - QKF0VRK34() (function in HMCFEI.SPR)
 - BROWSEITEM (procedure in HMCFEI.SPR)
 - HMCFEI.DBF (database)
 - HMCFE.DBF (database)
 - HMCF.DBF (database)
 - GET_HMCF... (procedure in HMCFE.SPR)
 - GET_HMCFE... (procedure in HMCFEI.SPR)
 - QKF0VRKGO() (function in HMCFEI.SPR)
 - QKF0VRKIQ() (function in HMCFEI.SPR)
 - REL() ... (function in HMCFE.SPR)
 - GET_HMCF... (procedure in HMCFE.SPR)
 - GET_HMCFE... (procedure in HMCFEI.SPR)
 - QKF0VRKMH() (function in HMCFEI.SPR)
 - QKF0VRKNV() (function in HMCFEI.SPR)
 - REL() ... (function in HMCFE.SPR)
 - GET_HMCF... (procedure in HMCFE.SPR)
 - GET_HMCFE... (procedure in HMCFEI.SPR)
 - QKF0VRKS5() (function in HMCFEI.SPR)
 - QKF0VRKTM() (function in HMCFEI.SPR)
 - GET_CF() ... (function in HMCFE.SPR)
 - GET_CFID... (procedure in HMCFE.SPR)
 - QKF0VRKWJ() (function in HMCFEI.SPR)
 - QKF0VRKXZ() (function in HMCFEI.SPR)
 - GET_CFE (procedure in HMCFEI.SPR)
 - HMCF.DBF (database)
 - CHOOSE.PRG...

```

      ERRMSG.PRG...
      GET_CFEID (procedure in HMCFEI.SPR)
        HMC.FDBF (database)
        IF.DBF (database)
        !EMPTY(X[1]).DBF (database)
      QKF0VRL1J() (function in HMCFEI.SPR)
      CHANGE... (procedure in HMCFEI.SPR)
      GET_HMCF... (procedure in HMCFEI.SPR)
      GET_HMCFE... (procedure in HMCFEI.SPR)
HMTAB.SPR
  HMUNIT.DBF (database)
  HMTAB.DBF (database)
  HMLU.PRG...
  DP.PRG...
  ERRMSG.PRG...
  GET_HMCOMID() (function in HMTAB.SPR)
    HMA.DBF (database)
    IF.DBF (database)
    !=.DBF (database)
    HMCOM.DBF (database)
  GET_HMLCN() ... (function in HMTAB.SPR)
  GET_HMWPID() ... (function in HMTAB.SPR)
  GET_HMETN() (function in HMTAB.SPR)
    HME.DBF (database)
  GET_CFAR() (function in HMTAB.SPR)
    HMC.DBF (database)
  GET_CFAR1() (function in HMTAB.SPR)
    HMC.DBF (database)
  GET_EI() (function in HMTAB.SPR)
    HMC.DBF (database)
  QKF0VRPCS() (function in HMTAB.SPR)
    GET_HMLCN() ... (function in HMTAB.SPR)
    GET_HMWPID() ... (function in HMTAB.SPR)
    GET_HMETN() ... (function in HMTAB.SPR)
    GET_CFAR() ... (function in HMTAB.SPR)
    GET_CFAR1() ... (function in HMTAB.SPR)
    GET_EI() ... (function in HMTAB.SPR)
  QKF0VRPYS() (function in HMTAB.SPR)
  QKF0VRQ1C() (function in HMTAB.SPR)
    HML.DBF (database)
    M.HMLCID=X[1] (database)
    GET_HMLC() ... (function in HMTAB.SPR)
  QKF0VRQ4T() (function in HMTAB.SPR)
  QKF0VRQ72() (function in HMTAB.SPR)
    HMW.DBF (database)
    M.HMWPID=X[1] (database)
    GET_HMWP() ... (function in HMTAB.SPR)
  QKF0VRQAL() (function in HMTAB.SPR)
  QKF0VRQC2() (function in HMTAB.SPR)
    HME.DBF (database)
    M.HMETID=X[1] (database)
    GET_HMET() (function in HMTAB.SPR)
      HME.DBF (database)
      CHOOSER.PRG...
      ERRMSG.PRG...
  QKF0VRQFR() (function in HMTAB.SPR)
  QKF0VRQH7() (function in HMTAB.SPR)
    HMC.DBF (database)
    GET_CF() ... (function in HMCFEI.SPR)

```


Section VI. Procedure and Function Summary. There are 26 files containing procedures in the HMLCCM system: HMINIT.PRG, HMAT.SPR, HMLC.SPR, HMWP.SPR, HMET.SPR, HMC.F.SPR, HMC.FE.SPR, HMC.FEI.SPR, HMTAB.SPR, BACKUP.PRG, HMCOMP.SPR, CHOOSER.PRG, HMLU.PRG, W_PRINT.SPR, HMSC.PRG, HMSCEN.PRG, HMSTEP.PRG, MEMOEDIT.PRG, DISPLAY.PRG, BWFACT.PRG, BSELECT.PRG, HMREF.PRG, HMMSDS.PRG, MEMOWIN.PRG, HMENU.MPR, and &OLDPROC.

1. HMINIT.PRG

Contains: MYHANDLER() (Params: none)
 Called by: HMINIT.PRG
 Called by: HMINIT.PRG
 Contains: _QUIT() (Params: none)
 Called by: HMENU.MPR

2. HMAT.SPR

Contains: CHANGE (Params: none)
 Called by: _QKF0VR394() (function in HMAT.SPR)
 Called by: _QKF0VR3EK() (function in HMAT.SPR)
 Called by: _QKF0VR3HM() (function in HMAT.SPR)
 Called by: _QKF0VR3ON() (function in HMAT.SPR)
 Called by: _QKF0VR3XH() (function in HMAT.SPR)
 Called by: _QKF0VR6VW() (function in HMLC.SPR)
 Called by: _QKF0VRBPA() (function in HMET.SPR)
 Called by: _QKF0VRL1J() (function in HMC.FEI.SPR)
 Called by: _QKF0VRQRG() (function in HMTAB.SPR)
 Called by: _QIP0VATWZ() (function in HMSTEP.PRG)
 Contains: NEWLINE() (Params: TEXT)
 Called by: HMVIEW() (function in HMREF.PRG)
 Contains: COMNAME() (Params: MKEY)
 Called by: HMAT.SPR
 Called by: _QKF0VR3HM() (function in HMAT.SPR)
 Called by: _QKF0VR3R9() (function in HMAT.SPR)
 Called by: _QKF0VR3XH() (function in HMAT.SPR)
 Contains: SAVECOM (Params: MKEY, MNAME)
 Called by: _QKF0VR3XH() (function in HMAT.SPR)
 Contains: _QKF0VR394() (Params: none)
 Called by: HMAT.SPR
 Calls: ERRMSG.PRG
 Calls: CHANGE (procedure in HMAT.SPR)
 Contains: _QKF0VR3EK() (Params: none)
 Called by: HMAT.SPR
 Calls: CHANGE (procedure in HMAT.SPR)
 Contains: _QKF0VR3HM() (Params: none)
 Called by: HMAT.SPR
 Calls: COMNAME() (function in HMAT.SPR)
 Calls: CHANGE (procedure in HMAT.SPR)
 Contains: _QKF0VR3KQ() (Params: none)
 Called by: HMAT.SPR
 Contains: _QKF0VR3ME() (Params: none)
 Called by: HMAT.SPR
 Contains: _QKF0VR3ON() (Params: none)
 Called by: HMAT.SPR
 Calls: CHANGE (procedure in HMAT.SPR)
 Contains: _QKF0VR3R9() (Params: none)
 Called by: HMAT.SPR
 Calls: COMNAME() (function in HMAT.SPR)

Contains: _QKF0VR3XH()	(Params: none)
Called by: HMAT.SPR	
Calls: ERRMSG.PRG	
Calls: CHANGE	(procedure in HMAT.SPR)
Calls: SAVECOM	(procedure in HMAT.SPR)
Calls: COMNAME()	(function in HMAT.SPR)

3. HMLC.SPR

Contains: CHANGE	(Params: none)
Called by: _QKF0VR394()	(function in HMAT.SPR)
Called by: _QKF0VR3EK()	(function in HMAT.SPR)
Called by: _QKF0VR3HM()	(function in HMAT.SPR)
Called by: _QKF0VR3ON()	(function in HMAT.SPR)
Called by: _QKF0VR3XH()	(function in HMAT.SPR)
Called by: _QKF0VR6VW()	(function in HMLC.SPR)
Called by: _QKF0VRBPA()	(function in HMET.SPR)
Called by: _QKF0VRL1J()	(function in HMCFEI.SPR)
Called by: _QKF0VRQRG()	(function in HMTAB.SPR)
Called by: _QIP0VATWZ()	(function in HMSTEP.PRG)
Contains: _QKF0VR6JX()	(Params: none)
Called by: HMLC.SPR	
Contains: _QKF0VR6TE()	(Params: none)
Called by: HMLC.SPR	
Calls: ERRMSG.PRG	
Contains: _QKF0VR6VW()	(Params: none)
Called by: HMLC.SPR	
Calls: CHANGE	(procedure in HMAT.SPR)

4. HMWP.SPR

Contains: CHANGE	(Params: none)
Called by: _QKF0VR394()	(function in HMAT.SPR)
Called by: _QKF0VR3EK()	(function in HMAT.SPR)
Called by: _QKF0VR3HM()	(function in HMAT.SPR)
Called by: _QKF0VR3ON()	(function in HMAT.SPR)
Called by: _QKF0VR3XH()	(function in HMAT.SPR)
Called by: _QKF0VR6VW()	(function in HMLC.SPR)
Called by: _QKF0VRBPA()	(function in HMET.SPR)
Called by: _QKF0VRL1J()	(function in HMCFEI.SPR)
Called by: _QKF0VRQRG()	(function in HMTAB.SPR)
Called by: _QIP0VATWZ()	(function in HMSTEP.PRG)
Contains: _QKF0VR92I()	(Params: none)
Called by: HMWP.SPR	
Contains: _QKF0VR99I()	(Params: none)
Called by: HMWP.SPR	
Calls: ERRMSG.PRG	
Contains: _QKF0VR9C5()	(Params: none)
Called by: HMWP.SPR	
Calls: ERRMSG.PRG	

5. HMET.SPR

Contains: CHANGE	(Params: none)
Called by: _QKF0VR394()	(function in HMAT.SPR)
Called by: _QKF0VR3EK()	(function in HMAT.SPR)
Called by: _QKF0VR3HM()	(function in HMAT.SPR)
Called by: _QKF0VR3ON()	(function in HMAT.SPR)
Called by: _QKF0VR3XH()	(function in HMAT.SPR)

Called by: _QKF0VR6VW()	(function in HMLC.SPR)
Called by: _QKF0VRBPA()	(function in HMET.SPR)
Called by: _QKF0VRL1J()	(function in HMCFEI.SPR)
Called by: _QKF0VRQRG()	(function in HMTAB.SPR)
Called by: _QIP0VATWZ()	(function in HMSTEP.PRG)
Contains: _QKF0VRBIT()	(Params: none)
Called by: HMET.SPR	
Contains: _QKF0VRBPA()	(Params: none)
Called by: HMET.SPR	
Calls: CHANGE	(procedure in HMAT.SPR)
Contains: _QKF0VRBT3()	(Params: none)
Called by: HMET.SPR	
Calls: ERRMSG.PRG	

6. HMC.F.SPR

Contains: CHANGE	(Params: none)
Called by: _QKF0VR394()	(function in HMAT.SPR)
Called by: _QKF0VR3EK()	(function in HMAT.SPR)
Called by: _QKF0VR3HM()	(function in HMAT.SPR)
Called by: _QKF0VR3ON()	(function in HMAT.SPR)
Called by: _QKF0VR3XH()	(function in HMAT.SPR)
Called by: _QKF0VR6VW()	(function in HMLC.SPR)
Called by: _QKF0VRBPA()	(function in HMET.SPR)
Called by: _QKF0VRL1J()	(function in HMCFEI.SPR)
Called by: _QKF0VRQRG()	(function in HMTAB.SPR)
Called by: _QIP0VATWZ()	(function in HMSTEP.PRG)
Contains: _QKF0VRDYQ()	(Params: none)
Called by: HMC.F.SPR	
Contains: _QKF0VRE5A()	(Params: none)
Called by: HMC.F.SPR	
Calls: ERRMSG.PRG	
Contains: _QKF0VRE7O()	(Params: none)
Called by: HMC.F.SPR	
Calls: ERRMSG.PRG	

7. HMC.FE.SPR

Contains: ESCPRESSED	(Params: none)
Contains: CHANGE	(Params: none)
Called by: _QKF0VR394()	(function in HMAT.SPR)
Called by: _QKF0VR3EK()	(function in HMAT.SPR)
Called by: _QKF0VR3HM()	(function in HMAT.SPR)
Called by: _QKF0VR3ON()	(function in HMAT.SPR)
Called by: _QKF0VR3XH()	(function in HMAT.SPR)
Called by: _QKF0VR6VW()	(function in HMLC.SPR)
Called by: _QKF0VRBPA()	(function in HMET.SPR)
Called by: _QKF0VRL1J()	(function in HMCFEI.SPR)
Called by: _QKF0VRQRG()	(function in HMTAB.SPR)
Called by: _QIP0VATWZ()	(function in HMSTEP.PRG)
Contains: GET_HMCF	(Params: M.HMCFID)
Called by: HMC.FE.SPR	
Called by: HMC.FE.I.SPR	
Called by: _QKF0VRGGZ()	(function in HMC.FE.SPR)
Called by: _QKF0VRGTP()	(function in HMC.FE.SPR)
Called by: _QKF0VRGYG()	(function in HMC.FE.SPR)
Called by: _QKF0VRH6D()	(function in HMC.FE.SPR)

Called by: _QKF0VRK34()	(function in HMCFEI.SPR)
Called by: _QKF0VRKIQ()	(function in HMCFEI.SPR)
Called by: _QKF0VRKRV()	(function in HMCFEI.SPR)
Called by: _QKF0VRL1J()	(function in HMCFEI.SPR)
Contains: GET_CF()	(Params: M.HMCF)
Called by: _QKF0VRH34()	(function in HMCFE.SPR)
Called by: _QKF0VRKTM()	(function in HMCFEI.SPR)
Called by: _QKF0VRQH7()	(function in HMTAB.SPR)
Calls: CHOOSER.PRG	
Calls: ERRMSG.PRG	
Contains: GET_CFID	(Params: M.ANSWR)
Called by: _QKF0VRH34()	(function in HMCFE.SPR)
Called by: _QKF0VRKTM()	(function in HMCFEI.SPR)
Contains: REL()	(Params: M.ID)
Called by: _QKF0VRGTP()	(function in HMCFE.SPR)
Called by: _QKF0VRGYG()	(function in HMCFE.SPR)
Called by: _QKF0VRKIQ()	(function in HMCFEI.SPR)
Called by: _QKF0VRKRV()	(function in HMCFEI.SPR)
Called by: _QKF0VRQKZ()	(function in HMTAB.SPR)
Contains: _QKF0VRGGZ()	(Params: none)
Called by: HMCFE.SPR	
Calls: GET_HMCF	(procedure in HMCFE.SPR)
Contains: _QKF0VRGRP()	(Params: none)
Called by: HMCFE.SPR	
Contains: _QKF0VRGTP()	(Params: none)
Called by: HMCFE.SPR	
Calls: REL()	(function in HMCFE.SPR)
Calls: GET_HMCF	(procedure in HMCFE.SPR)
Contains: _QKF0VRGX4()	(Params: none)
Called by: HMCFE.SPR	
Contains: _QKF0VRGYG()	(Params: none)
Called by: HMCFE.SPR	
Calls: REL()	(function in HMCFE.SPR)
Calls: GET_HMCF	(procedure in HMCFE.SPR)
Contains: _QKF0VRH1G()	(Params: none)
Called by: HMCFE.SPR	
Contains: _QKF0VRH34()	(Params: none)
Called by: HMCFE.SPR	
Calls: GET_CF()	(function in HMCFE.SPR)
Calls: GET_CFID	(procedure in HMCFE.SPR)
Contains: _QKF0VRH6D()	(Params: none)
Called by: HMCFE.SPR	
Calls: GET_HMCF	(procedure in HMCFE.SPR)

8. HMCFEI.SPR

Contains: ESCPRESSED	(Params: none)
Contains: CHANGE	(Params: none)
Called by: _QKF0VR394()	(function in HMAT.SPR)
Called by: _QKF0VR3EK()	(function in HMAT.SPR)
Called by: _QKF0VR3HM()	(function in HMAT.SPR)
Called by: _QKF0VR3ON()	(function in HMAT.SPR)
Called by: _QKF0VR3XH()	(function in HMAT.SPR)
Called by: _QKF0VR6VW()	(function in HMLC.SPR)
Called by: _QKF0VRBPA()	(function in HMET.SPR)
Called by: _QKF0VRL1J()	(function in HMCFEI.SPR)
Called by: _QKF0VRQRG()	(function in HMTAB.SPR)
Called by: _QIP0VATWZ()	(function in HMSTEP.PRG)
Contains: GET_HMCF	(Params: M.HMCFID)

Called by: HMCFE.SPR	
Called by: HMCFEI.SPR	
Called by: _QKF0VRGGZ()	(function in HMCFE.SPR)
Called by: _QKF0VRGTP()	(function in HMCFE.SPR)
Called by: _QKF0VRGYG()	(function in HMCFE.SPR)
Called by: _QKF0VRH6D()	(function in HMCFE.SPR)
Called by: _QKF0VRK34()	(function in HMCFEI.SPR)
Called by: _QKF0VRKIQ()	(function in HMCFEI.SPR)
Called by: _QKF0VRKNV()	(function in HMCFEI.SPR)
Called by: _QKF0VRL1J()	(function in HMCFEI.SPR)
Contains: GET_CF()	(Params: M.HMCF)
Called by: _QKF0VRH34()	(function in HMCFE.SPR)
Called by: _QKF0VRKTM()	(function in HMCFEI.SPR)
Called by: _QKF0VRQH7()	(function in HMTAB.SPR)
Calls: CHOOSER.PRG	
Calls: ERRMSG.PRG	
Contains: GET_CFID	(Params: M.ANSWR)
Called by: _QKF0VRH34()	(function in HMCFE.SPR)
Called by: _QKF0VRKTM()	(function in HMCFEI.SPR)
Contains: GET_HMCFE	(Params: M.HMCFEID)
Called by: HMCFEI.SPR	
Called by: _QKF0VRK34()	(function in HMCFEI.SPR)
Called by: _QKF0VRKIQ()	(function in HMCFEI.SPR)
Called by: _QKF0VRKNV()	(function in HMCFEI.SPR)
Called by: _QKF0VRL1J()	(function in HMCFEI.SPR)
Contains: GET_CFE	(Params: M.HMCFE)
Called by: _QKF0VRKXZ()	(function in HMCFEI.SPR)
Calls: CHOOSER.PRG	
Calls: ERRMSG.PRG	
Contains: GET_CFEID	(Params: M.HMCFE)
Called by: _QKF0VRKXZ()	(function in HMCFEI.SPR)
Contains: REL()	(Params: M.ID)
Called by: _QKF0VRGTP()	(function in HMCFE.SPR)
Called by: _QKF0VRGYG()	(function in HMCFE.SPR)
Called by: _QKF0VRKIQ()	(function in HMCFEI.SPR)
Called by: _QKF0VRKNV()	(function in HMCFEI.SPR)
Called by: _QKF0VRQKZ()	(function in HMTAB.SPR)
Contains: BROWSEITEM	(Params: M.ID)
Called by: _QKF0VRK34()	(function in HMCFEI.SPR)
Contains: _QKF0VRK34()	(Params: none)
Called by: HMCFEI.SPR	
Calls: BROWSEITEM	(procedure in HMCFEI.SPR)
Calls: GET_HMCF	(procedure in HMCFE.SPR)
Calls: GET_HMCFE	(procedure in HMCFEI.SPR)
Contains: _QKF0VRKGO()	(Params: none)
Called by: HMCFEI.SPR	
Contains: _QKF0VRKIQ()	(Params: none)
Called by: HMCFEI.SPR	
Calls: REL()	(function in HMCFE.SPR)
Calls: GET_HMCF	(procedure in HMCFE.SPR)
Calls: GET_HMCFE	(procedure in HMCFEI.SPR)
Contains: _QKF0VRKMH()	(Params: none)
Called by: HMCFEI.SPR	
Contains: _QKF0VRKNV()	(Params: none)
Called by: HMCFEI.SPR	
Calls: REL()	(function in HMCFE.SPR)
Calls: GET_HMCF	(procedure in HMCFE.SPR)
Calls: GET_HMCFE	(procedure in HMCFEI.SPR)
Contains: _QKF0VRKS5()	(Params: none)

Called by: HMCFEI.SPR	
Contains: _QKF0VRKTM()	(Params: none)
Called by: HMCFEI.SPR	
Calls: GET_CF()	(function in HMCFE.SPR)
Calls: GET_CFID	(procedure in HMCFE.SPR)
Contains: _QKF0VRKWJ()	(Params: none)
Called by: HMCFEI.SPR	
Contains: _QKF0VRKXZ()	(Params: none)
Called by: HMCFEI.SPR	
Calls: GET_CFE	(procedure in HMCFEI.SPR)
Calls: GET_CFEID	(procedure in HMCFEI.SPR)
Contains: _QKF0VRL1J()	(Params: none)
Called by: HMCFEI.SPR	
Calls: CHANGE	(procedure in HMCFEI.SPR)
Calls: GET_HMCF	(procedure in HMCFEI.SPR)
Calls: GET_HMCFE	(procedure in HMCFEI.SPR)

9. HMTAB.SPR

Contains: ESCPRESSED	(Params: none)
Contains: CHANGE	(Params: none)
Called by: _QKF0VR394()	(function in HMTAB.SPR)
Called by: _QKF0VR3EK()	(function in HMTAB.SPR)
Called by: _QKF0VR3HM()	(function in HMTAB.SPR)
Called by: _QKF0VR3ON()	(function in HMTAB.SPR)
Called by: _QKF0VR3XH()	(function in HMTAB.SPR)
Called by: _QKF0VR6VW()	(function in HMLC.SPR)
Called by: _QKF0VRBPA()	(function in HMET.SPR)
Called by: _QKF0VRL1J()	(function in HMCFEI.SPR)
Called by: _QKF0VRQRG()	(function in HMTAB.SPR)
Called by: _QIP0VATWZ()	(function in HMSTEP.PRG)
Contains: GET_HMCOMID()	(Params: M.HMATID)
Called by: HMTAB.SPR	
Contains: GET_HMLC()	(Params: M.HMLC)
Called by: _QKF0VRQ1C()	(function in HMTAB.SPR)
Called by: _QIP0VAT9X()	(function in HMSTEP.PRG)
Calls: CHOOSER.PRG	
Calls: ERRMSG.PRG	
Contains: GET_HMLCN()	(Params: M.ID)
Called by: HMTAB.SPR	
Called by: _QKF0VRPCS()	(function in HMTAB.SPR)
Called by: _QKF0VRQRG()	(function in HMTAB.SPR)
Called by: INITVAR	(procedure in HMSTEP.PRG)
Contains: GET_HMWP()	(Params: M.MATCH)
Called by: _QKF0VRQ72()	(function in HMTAB.SPR)
Called by: _QIP0VATLC()	(function in HMSTEP.PRG)
Calls: CHOOSER.PRG	
Calls: ERRMSG.PRG	
Contains: GET_HMWPN()	(Params: M.ID)
Called by: HMTAB.SPR	
Called by: _QKF0VRPCS()	(function in HMTAB.SPR)
Called by: _QKF0VRQRG()	(function in HMTAB.SPR)
Called by: INITVAR	(procedure in HMSTEP.PRG)
Contains: GET_HMET()	(Params: M.HMET)
Called by: _QKF0VRQC2()	(function in HMTAB.SPR)
Calls: CHOOSER.PRG	
Calls: ERRMSG.PRG	
Contains: GET_HMETID()	(Params: M.HMET)
Contains: GET_HMETN()	(Params: M.HMETID)

Called by: HMTAB.SPR	
Called by: _QKF0VRPCS()	(function in HMTAB.SPR)
Called by: _QKF0VRQRG()	(function in HMTAB.SPR)
Contains: GET_CFAR()	(Params: M.HMCFID)
Called by: HMTAB.SPR	
Called by: _QKF0VRPCS()	(function in HMTAB.SPR)
Called by: _QKF0VRQRG()	(function in HMTAB.SPR)
Contains: GET_CFAR1()	(Params: M.HMCFEID)
Called by: HMTAB.SPR	
Called by: _QKF0VRPCS()	(function in HMTAB.SPR)
Called by: _QKF0VRQRG()	(function in HMTAB.SPR)
Contains: GET_CF()	(Params: M.HMCF)
Called by: _QKF0VRH34()	(function in HMCFE.SPR)
Called by: _QKF0VRKTM()	(function in HMCFEI.SPR)
Called by: _QKF0VRQH7()	(function in HMTAB.SPR)
Calls: CHOOSER.PRG	
Calls: ERRMSG.PRG	
Contains: GET_HMCFID()	(Params: M.HMCF)
Called by: _QKF0VRQKZ()	(function in HMTAB.SPR)
Contains: GET_CF1()	(Params: M.HMCFE)
Calls: CHOOSER.PRG	
Calls: ERRMSG.PRG	
Contains: GET_EID()	(Params: M.HMCFE)
Called by: _QKF0VRQKZ()	(function in HMTAB.SPR)
Contains: GET_EI()	(Params: M.HMCFEIID)
Called by: HMTAB.SPR	
Called by: _QKF0VRPCS()	(function in HMTAB.SPR)
Called by: _QKF0VRQRG()	(function in HMTAB.SPR)
Contains: GET_EIID()	(Params: M.HMCFEI)
Called by: _QKF0VRQKZ()	(function in HMTAB.SPR)
Contains: REL()	(Params: M.ID)
Called by: _QKF0VRGTP()	(function in HMCFE.SPR)
Called by: _QKF0VRGYG()	(function in HMCFE.SPR)
Called by: _QKF0VRKIQ()	(function in HMCFEI.SPR)
Called by: _QKF0VRKNV()	(function in HMCFEI.SPR)
Called by: _QKF0VRQKZ()	(function in HMTAB.SPR)
Contains: _QKF0VRPCS()	(Params: none)
Called by: HMTAB.SPR	
Calls: GET_HMLCN()	(function in HMTAB.SPR)
Calls: GET_HMWP()	(function in HMTAB.SPR)
Calls: GET_HMETN()	(function in HMTAB.SPR)
Calls: GET_CFAR()	(function in HMTAB.SPR)
Calls: GET_CFAR1()	(function in HMTAB.SPR)
Calls: GET_EI()	(function in HMTAB.SPR)
Contains: _QKF0VRPYS()	(Params: none)
Called by: HMTAB.SPR	
Contains: _QKF0VRQ1C()	(Params: none)
Called by: HMTAB.SPR	
Calls: GET_HMLC()	(function in HMTAB.SPR)
Contains: _QKF0VRQ4T()	(Params: none)
Called by: HMTAB.SPR	
Contains: _QKF0VRQ72()	(Params: none)
Called by: HMTAB.SPR	
Calls: GET_HMWP()	(function in HMTAB.SPR)
Contains: _QKF0VRQAL()	(Params: none)
Called by: HMTAB.SPR	
Contains: _QKF0VRQC2()	(Params: none)
Called by: HMTAB.SPR	
Calls: GET_HMET()	(function in HMTAB.SPR)

Contains: _QKFOVRQFR()	(Params: none)
Called by: HMTAB.SPR	
Contains: _QKFOVRQH7()	(Params: none)
Called by: HMTAB.SPR	
Calls: GET_CF()	(function in HMCFE.SPR)
Contains: _QKFOVRQKZ()	(Params: none)
Called by: HMTAB.SPR	
Calls: REL()	(function in HMCFE.SPR)
Calls: GET_HMCFID()	(function in HMTAB.SPR)
Calls: GET_EID()	(function in HMTAB.SPR)
Calls: GET_EIID()	(function in HMTAB.SPR)
Contains: _QKFOVRQRG()	(Params: none)
Called by: HMTAB.SPR	
Calls: CHANGE	(procedure in HMTAB.SPR)
Calls: GET_HMLCN()	(function in HMTAB.SPR)
Calls: GET_HMWPB()	(function in HMTAB.SPR)
Calls: GET_HMETN()	(function in HMTAB.SPR)
Calls: GET_CFBAR()	(function in HMTAB.SPR)
Calls: GET_CFBAR1()	(function in HMTAB.SPR)
Calls: GET_EI()	(function in HMTAB.SPR)

10. **BACKUP.PRG**

Contains: ERRHAND	(Params: none)
Called by: ERRMSG.PRG	
Contains: STOP	(Params: none)
Called by: ERRMSG.PRG	

11. **HMCOMP.SPR**

Contains: OPEN	(Params: none)
Called by: HMCOMP.SPR	
Calls: OPENFILE()	(function in HMSC.PRG)
Calls: CANCEL	(procedure in HMCOMP.SPR)
Contains: DELFILE()	(Params: FILE)
Called by: HMCOMP.SPR	
Called by: CALCULATE()	(function in HMCOMP.SPR)
Called by: SELWT()	(function in HMCOMP.SPR)
Called by: BSTEP()	(function in HMCOMP.SPR)
Called by: BFACT()	(function in HMCOMP.SPR)
Called by: BWSTEP()	(function in HMCOMP.SPR)
Called by: BPHASE()	(function in HMCOMP.SPR)
Contains: CANCEL	(Params: SUCCESS)
Called by: HMSTEP.PRG	
Called by: OPEN	(procedure in HMCOMP.SPR)
Calls: ERRMSG.PRG	
Contains: GET_HMARRAY	(Params: none)
Called by: HMCOMP.SPR	
Calls: GET_TABLE	(procedure in HMCOMP.SPR)
Contains: GET_TABLE	(Params: COMID, LCID, WPID)
Called by: GET_HMARRAY	(procedure in HMCOMP.SPR)
Contains: COMPUTE	(Params: none)
Called by: _QLE0L6CB0()	(function in HMCOMP.SPR)
Calls: SUBCOMPUT	(procedure in HMCOMP.SPR)
Calls: DISPLAY.PRG	
Contains: SUBCOMPUT	(Params: M.IHMATNAME, M.IHMATID)
Called by: COMPUTE	(procedure in HMCOMP.SPR)
Calls: COMPUTSTEP()	(function in HMCOMP.SPR)

Calls: COMPUTFACT()	(function in HMCOMP.SPR)
Calls: COMPUTWFACT()	(function in HMCOMP.SPR)
Calls: COMPUTPHASE()	(function in HMCOMP.SPR)
Calls: BSTEP()	(function in HMCOMP.SPR)
Calls: DP.PRG	
Calls: BFACT()	(function in HMCOMP.SPR)
Calls: BWSTEP()	(function in HMCOMP.SPR)
Calls: BPHASE()	(function in HMCOMP.SPR)
Calls: GETSTR()	(function in HMCOMP.SPR)
Contains: HMBROWSE	(Params: none)
Called by: _QLE0L6CB0()	(function in HMCOMP.SPR)
Contains: COMPUTSTEP()	(Params: COM, HMAT, LCPHASE, WP, SN,
SD, SQ)	
Called by: SUBCOMPUT	(procedure in HMCOMP.SPR)
Contains: FACTNAME()	(Params: MFACTID)
Contains: COMPUTFACT()	(Params: M.FACTID, M.HMATID)
Called by: SUBCOMPUT	(procedure in HMCOMP.SPR)
Contains: COMPUTWFACT()	(Params: M.FACTID, M.HMATID)
Called by: SUBCOMPUT	(procedure in HMCOMP.SPR)
Contains: COMPUTPHASE()	(Params: M.PHASEID, M.HMATID)
Called by: SUBCOMPUT	(procedure in HMCOMP.SPR)
Contains: CALCULATE()	(Params: SN, SD, SQ)
Calls: DELFILE()	(function in HMCOMP.SPR)
Contains: RESETWT()	(Params: MATID, LCPHASE, WP)
Called by: BSTEP()	(function in HMCOMP.SPR)
Called by: BFACT()	(function in HMCOMP.SPR)
Called by: BWSTEP()	(function in HMCOMP.SPR)
Called by: BPHASE()	(function in HMCOMP.SPR)
Calls: SELWT()	(function in HMCOMP.SPR)
Contains: SELWT()	(Params: MATID, LCPHASE, WP, METID,
CFID, CFEID)	
Called by: RESETWT()	(function in HMCOMP.SPR)
Calls: DELFILE()	(function in HMCOMP.SPR)
Contains: SETUPBOOT()	(Params: none)
Called by: _QLE0L6CB0()	(function in HMCOMP.SPR)
Calls: BSELECT.PRG	
Calls: ERRMSG.PRG	
Calls: BWFACT.PRG	
Calls: DP.PRG	
Contains: BSTEP()	(Params: MAT, LCPHASE, WP, SN, SD, SQ)
Called by: SUBCOMPUT	(procedure in HMCOMP.SPR)
Calls: RESETWT()	(function in HMCOMP.SPR)
Calls: DELFILE()	(function in HMCOMP.SPR)
Contains: BFACT()	(Params: M.FACTID, M.HMATID)
Called by: SUBCOMPUT	(procedure in HMCOMP.SPR)
Calls: RESETWT()	(function in HMCOMP.SPR)
Calls: DELFILE()	(function in HMCOMP.SPR)
Contains: BWSTEP()	(Params: M.FACTID, M.STEPID, M.HMATID)
Called by: SUBCOMPUT	(procedure in HMCOMP.SPR)
Calls: RESETWT()	(function in HMCOMP.SPR)
Calls: DELFILE()	(function in HMCOMP.SPR)
Contains: BPHASE()	(Params: M.PHASEID, M.HMATID)
Called by: SUBCOMPUT	(procedure in HMCOMP.SPR)
Calls: RESETWT()	(function in HMCOMP.SPR)
Calls: DELFILE()	(function in HMCOMP.SPR)
Contains: GETSTR()	(Params: HMATNAME, SCENTOTAL,
STEPTOTAL, FACTOTAL, WFACTOTAL, PHASETOTAL, BOOTSTRAP)	
Called by: SUBCOMPUT	(procedure in HMCOMP.SPR)
Calls: DP.PRG	

Contains: _QLE0L6B8T()	(Params: none)
Called by: HMCOMP.SPR	
Contains: _QLE0L6BIV()	(Params: none)
Called by: HMCOMP.SPR	
Contains: _QLE0L6BR7()	(Params: none)
Called by: HMCOMP.SPR	
Contains: _QLE0L6BYN()	(Params: none)
Called by: HMCOMP.SPR	
Contains: _QLE0L6C5N()	(Params: none)
Called by: HMCOMP.SPR	
Contains: _QLE0L6CB0()	(Params: none)
Called by: HMCOMP.SPR	
Calls: SETUPBOOT()	(function in HMCOMP.SPR)
Calls: POPUPSHOW()	(function in HMSC.PRG)
Calls: COMPUTE	(procedure in HMCOMP.SPR)
Calls: POPUPHIDE()	(function in HMSC.PRG)
Calls: HMBROWSE	(procedure in HMCOMP.SPR)
12. CHOOSER.PRG	
Contains: _Q120IDTF()	(Params: none)
Called by: CHOOSER.PRG	
Contains: _Q120IDG17()	(Params: none)
Called by: CHOOSER.PRG	
Contains: _Q120IDG86()	(Params: none)
Called by: CHOOSER.PRG	
13. HMLU.PRG	
Contains: GET_HMNAME	(Params: M.NAME)
Called by: _Q7D001EI8()	(function in HMLU.PRG)
Calls: CHOOSER.PRG	
Calls: ERRMSG.PRG	
Contains: _Q7D001EI8()	(Params: none)
Called by: HMLU.PRG	
Calls: GET_HMNAME	(procedure in HMLU.PRG)
Calls: DP.PRG	
Contains: _Q7D001EUB()	(Params: none)
Called by: HMLU.PRG	
14. W_PRINT.SPR	
Contains: PRINTFILE()	(Params: none)
Called by: _QKF0VS4PH()	(function in W_PRINT.SPR)
Contains: _QKF0VS4NC()	(Params: none)
Called by: W_PRINT.SPR	
Contains: _QKF0VS4PH()	(Params: none)
Called by: W_PRINT.SPR	
Calls: PRINTFILE()	(function in W_PRINT.SPR)
Contains: _QKF0VS4RM()	(Params: none)
Called by: W_PRINT.SPR	
15. HMSC.PRG	
Contains: CLOSEFILE	(Params: none)
Called by: HMSC.PRG	
Contains: DEL_HMSC()	(Params: ID)
Called by: HMSC.PRG	
Calls: POPUPSHOW()	(function in HMSC.PRG)

Calls: OPENFILE()	(function in HMSC.PRG)
Calls: POPUPHIDE()	(function in HMSC.PRG)
Contains: FILEFIND()	(Params: MFILENAME)
Called by: OPENFILE()	(function in HMSC.PRG)
Calls: DP.PRG	
Contains: OPENFILE()	(Params: FILE)
Called by: HMSTEP.PRG	
Called by: HMREF.PRG	
Called by: HMMSDS.PRG	
Called by: OPEN	(procedure in HMSCOMP.SPR)
Called by: DEL_HMSC()	(function in HMSC.PRG)
Calls: FILEFIND()	(function in HMSC.PRG)
Contains: POPUPSHOW()	(Params: ERRSTR)
Called by: _QLE0L6CB0()	(function in HMSCOMP.SPR)
Called by: DEL_HMSC()	(function in HMSC.PRG)
Contains: POPUPHIDE()	(Params: W)
Called by: _QLE0L6CB0()	(function in HMSCOMP.SPR)
Called by: DEL_HMSC()	(function in HMSC.PRG)
16. HMSCEN.PRG	
Contains: GET_HMSC()	(Params: NAME)
Called by: _Q8K0INGJH()	(function in HMSCEN.PRG)
Called by: _Q8K0INGTV()	(function in HMSCEN.PRG)
Calls: CHOOSER.PRG	
Contains: MEVENT	(Params: MNEW)
Called by: _Q8K0INGJH()	(function in HMSCEN.PRG)
Called by: _Q8K0INGTV()	(function in HMSCEN.PRG)
Called by: _Q8K0INH BV()	(function in HMSCEN.PRG)
Contains: _Q8K0INGJH()	(Params: none)
Called by: HMSCEN.PRG	
Calls: GET_HMSC()	(function in HMSCEN.PRG)
Calls: MEVENT	(procedure in HMSCEN.PRG)
Calls: DP.PRG	
Contains: _Q8K0INGTV()	(Params: none)
Called by: HMSCEN.PRG	
Calls: ERRMSG.PRG	
Calls: GET_HMSC()	(function in HMSCEN.PRG)
Calls: MEVENT	(procedure in HMSCEN.PRG)
Calls: DP.PRG	
Contains: _Q8K0INH BV()	(Params: none)
Called by: HMSCEN.PRG	
Calls: MEVENT	(procedure in HMSCEN.PRG)
17. HMSTEP.PRG	
Contains: ESCPRESSED	(Params: none)
Contains: INITVAR	(Params: none)
Called by: HMSTEP.PRG	
Called by: _QIPOVASCN()	(function in HMSTEP.PRG)
Called by: _QIPOVATWZ()	(function in HMSTEP.PRG)
Calls: GET_HMATN()	(function in HMSTEP.PRG)
Calls: GET_HMLCN()	(function in HMTAB.SPR)
Calls: GET_HMWP N()	(function in HMTAB.SPR)
Contains: ADDOPTION	(Params: none)
Called by: _QIPOVASCN()	(function in HMSTEP.PRG)
Called by: _QIPOVATWZ()	(function in HMSTEP.PRG)
Contains: CHANGE	(Params: none)
Called by: _QKF0VR394()	(function in HMAT.SPR)

Called by: _QKF0VR3EK()	(function in HMAT.SPR)
Called by: _QKF0VR3HM()	(function in HMAT.SPR)
Called by: _QKF0VR3ON()	(function in HMAT.SPR)
Called by: _QKF0VR3XH()	(function in HMAT.SPR)
Called by: _QKF0VR6VW()	(function in HMLC.SPR)
Called by: _QKF0VRBPA()	(function in HMET.SPR)
Called by: _QKF0VRL1J()	(function in HMCFEI.SPR)
Called by: _QKF0VRQRG()	(function in HMTAB.SPR)
Called by: _QIPOVATWZ()	(function in HMSTEP.PRG)
Contains: GET_HMAT()	(Params: NAME)
Called by: _QIPOVASYA()	(function in HMSTEP.PRG)
Calls: CHOOSER.PRG	
Calls: ERRMSG.PRG	
Contains: GET_HMATN()	(Params: M.ID)
Called by: INITVAR	(procedure in HMSTEP.PRG)
Contains: GET_HMLC()	(Params: M.HMLC)
Called by: _QKF0VRQ1C()	(function in HMTAB.SPR)
Called by: _QIPOVAT9X()	(function in HMSTEP.PRG)
Calls: CHOOSER.PRG	
Calls: ERRMSG.PRG	
Contains: GET_HMLCN()	(Params: M.ID)
Called by: HMTAB.SPR	
Called by: _QKF0VRPCS()	(function in HMTAB.SPR)
Called by: _QKF0VRQRG()	(function in HMTAB.SPR)
Called by: INITVAR	(procedure in HMSTEP.PRG)
Contains: GET_HMWP()	(Params: M.MATCH)
Called by: _QKF0VRQ72()	(function in HMTAB.SPR)
Called by: _QIPOVATLC()	(function in HMSTEP.PRG)
Calls: CHOOSER.PRG	
Calls: ERRMSG.PRG	
Contains: GET_HMWPN()	(Params: M.ID)
Called by: HMTAB.SPR	
Called by: _QKF0VRPCS()	(function in HMTAB.SPR)
Called by: _QKF0VRQRG()	(function in HMTAB.SPR)
Called by: INITVAR	(procedure in HMSTEP.PRG)
Contains: GET_HMUNIT	(Params: none)
Called by: _QIPOVASYA()	(function in HMSTEP.PRG)
Called by: _QIPOVAT9X()	(function in HMSTEP.PRG)
Called by: _QIPOVATLC()	(function in HMSTEP.PRG)
Called by: DATACHECK()	(function in HMSTEP.PRG)
Calls: ERRMSG.PRG	
Contains: SCSAVE	(Params: none)
Called by: _QIPOVASCN()	(function in HMSTEP.PRG)
Contains: DATACHECK()	(Params: none)
Called by: _QIPOVATWZ()	(function in HMSTEP.PRG)
Calls: ERRMSG.PRG	
Calls: GET_HMUNIT	(procedure in HMSTEP.PRG)
Contains: _QIPOVASCN()	(Params: none)
Called by: HMSTEP.PRG	
Calls: YESNO.PRG	
Calls: INITVAR	(procedure in HMSTEP.PRG)
Calls: ADDOPTION	(procedure in HMSTEP.PRG)
Calls: SCSAVE	(procedure in HMSTEP.PRG)
Contains: _QIPOVASVS()	(Params: none)
Called by: HMSTEP.PRG	
Contains: _QIPOVASYA()	(Params: none)
Called by: HMSTEP.PRG	
Calls: GET_HMAT()	(function in HMSTEP.PRG)
Calls: DP.PRG	

Calls: GET_HMUNIT	(procedure in HMSTEP.PRG)
Contains: _QIPOVAT7G()	(Params: none)
Called by: HMSTEP.PRG	
Contains: _QIPOVAT9X()	(Params: none)
Called by: HMSTEP.PRG	
Calls: GET_HMLC()	(function in HMTAB.SPR)
Calls: DP.PRG	
Calls: GET_HMUNIT	(procedure in HMSTEP.PRG)
Contains: _QIPOVATIU()	(Params: none)
Called by: HMSTEP.PRG	
Contains: _QIPOVATLC()	(Params: none)
Called by: HMSTEP.PRG	
Calls: GET_HMWP()	(function in HMTAB.SPR)
Calls: DP.PRG	
Calls: GET_HMUNIT	(procedure in HMSTEP.PRG)
Contains: _QIPOVATWZ()	(Params: none)
Called by: HMSTEP.PRG	
Calls: DATACHECK()	(function in HMSTEP.PRG)
Calls: CHANGE	(procedure in HMAT.SPR)
Calls: YESNO.PRG	
Calls: INITVAR	(procedure in HMSTEP.PRG)
Calls: ADDOPTION	(procedure in HMSTEP.PRG)
Contains: _QIPOVAUD2()	(Params: none)
Called by: HMSTEP.PRG	
18. MEMOEDIT.PRG	
Contains: _Q8L0LJ6F0()	(Params: none)
Called by: MEMOEDIT.PRG	
19. DISPLAY.PRG	
Contains: _QIN0M9E2J()	(Params: none)
Called by: DISPLAY.PRG	
Calls: W_PRINT.SPR	
20. BWFACT.PRG	
Contains: _Q8Q0NQKA6()	(Params: none)
Called by: BWFACT.PRG	
21. BSELECT.PRG	
Contains: _Q8Q0N6SBM()	(Params: none)
Called by: BSELECT.PRG	
22. HMREF.PRG	
Contains: HMVIEW()	(Params: NAME)
Called by: HMREF.PRG	
Called by: HMMSDS.PRG	
Calls: NEWLINE()	(function in HMAT.SPR)
Calls: MEMOWIN.PRG	
Contains: NEWLINE()	(Params: TEXT)
Called by: HMVIEW()	(function in HMREF.PRG)
23. HMMSDS.PRG	
Contains: HMVIEW()	(Params: NAME)

Called by: HMREF.PRG
Called by: HMMSDS.PRG
Calls: NEWLINE()
Calls: MEMOWIN.PRG (function in HMAT.SPR)

24. **MEMOWIN.PRG**

Contains: _QIG0IXOXD() (Params: none)
Called by: MEMOWIN.PRG
Calls: W_PRINT.SPR
Contains: _QIG0IXPJO() (Params: none)
Called by: MEMOWIN.PRG
Calls: W_PRINT.SPR

25. **HMENU.MPR**

Contains: _QKF0VR0CN (Params: none)
Called by: HMENU.MPR
Calls: BACKUP.PRG
Contains: _QKF0VR0FQ (Params: none)
Called by: HMENU.MPR
Calls: ERRMSG.PRG

26. **&OLDPROC**

Temporary Variable

Section VII. Program Source Code


```

1  * *****
=> *****
2  * Procedure file: C:\HMLCCM\WORK\BACKUP.PRG
3  *
4  *
5  * System: Hazardous Material Life-Cycle Cost Model
6  * Author: Naval Health Research Center
7  * Copyright (c) 1993, Naval Health Research Center
8  * Last modified: 12/01/93 14:51
9  *
10 * Procs & Fncts: ERRHAND
11 * : STOP
12 *
13 * Set by: _OLEOVCDQC (procedure in HMENU.MPR)
14 *
15 * Calls: ERRMSG.PRG
16 *
17 * Documented 12/02/93 at 10:50 FoxDoc version 2.10f
18 * *****
=> *****
19 set talk off
20 =errmsg("Insert diskette in drive A and hit RETURN")
21 =errmsg("Please wait.",1)
22
23 ! pkzip a:\hmbakup.zip *.dbf
24 ! pkzip -u a:\hmbakup.zip *.cdx
25 ! pkzip -u a:\hmbakup.zip *.fpt
26
27 =errmsg("DONE.",1)
28
29 clear
30 return
31
32 *****
40 * *****
=> *****
41 *
42 * Procedure: ERRHAND
43 *
44 * Calls: ERRMSG.PRG
45 *
46 * *****
=> *****
40 procedure errhand
41 =errmsg("BACKUP CRASHED CHECK PARAMETERS AND TRY AGAIN.")
42 return
43 quit
44
45 *****
=> *****
54 * *****
=> *****
55 *
56 * Procedure: STOP
57 *
58 * Calls: EMPTY() (function in ?)
59 * : ERROR() (function in ?)
60 * : ERRMSG.PRG
61 *
62 * *****
=> *****
54 procedure stop
55 if empty(error())
56 =errmsg("Please check drive A")
57 retry
58 endif
59 return
60 * EOF: BACKUP.ACT

```

```

1  * *****
2  *
3  * Procedure file: C:\HAZMAT\GHW\WORK\BIGCHARS.PRG
4  *
5  * System: Hazardous Material Life-Cycle Cost Model
6  * Author: Naval Health Research Center
7  * Copyright (c) 1993, Naval Health Research Center
8  * Last modified: 09/13/93 8:43
9  *
10 * Set by: HMINIT.PRG
11 * : HMINIT.PRG
12 *
13 * Documented 12/01/93 at 11:31 FoxDoc version 2.10f
14 * *****
15 *
16 * FUNCTION: BIGCHARS
17 * PURPOSE: PRINT HEADER
18 * RETURN: None.
19 *
20 * PARAMETERS:
21 * startx x-coordinate to start (sugg: 6)
22 * starty y-coordinate to start (sugg: 17)
23 * instrng title to be blown up
24 * o_t waiting time (sugg: 6)
25 * REFERENCE: C:\FOXPRO2\GOODIES\DEMO\PRGS\BIGCHARS.PRG
26 * SIDE EFFECTS: Not known.
27 * CREATED BY: Anh Le 18 JUN 92
28 * MODIFIED: AL 23 JUN 92
29 *
30 * *****
31 * PARAMETER startx,starty,instrng,o_t
32 *
33 * IF PARAMETERS() = 3
34 * o_t = 6
35 * ENDIF
36
37 x_coor = startx
38 y_coor = starty
39 instrng = UPPER(instrng)
40 len_str = LEN(instrng)
41
42 DO CASE
43 CASE (len_str = 5)
44 starty = starty
45 CASE (len_str = 3)
46 starty = starty + 9
47 CASE (len_str < 5)
48 starty = starty + 3
49 CASE (len_str = 6)
50 starty = starty - 3
51 CASE (len_str = 7)
52 starty = starty - 6
53 OTHERWISE
54 starty = starty - starty+4
55 ENDCASE
56
57 SET BLINK OFF
58 CLEAR
59
60 IF startx > 20
61 RETURN
62 ENDIF
63
64 curentry = starty

```

```

65 curlet = 1
66
67 IF "HOMO" $ UPPER(SYS(2006))
68 colorvar = "n/n"
69 ELSE
70 colorvar = "n/b"
71 ENDIF
72 CLEAR
73
74 DO WHILE curlet <= LEN(instrng) AND curentry < 89
75 DO CASE
76 CASE SUBSTR(instrng,curlet,1) = " "
77 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) :
78 FROM startx,curentry TO startx + 3,curentry + 5 ;
79 NONE NOSHADOW
80 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
81 curentry = curentry + 6
82 CASE SUBSTR(instrng,curlet,1) = "t"
83 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) :
84 FROM startx,curentry TO startx + 5,curentry + 5 ;
85 NONE NOSHADOW
86 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
87 curentry = curentry + 6
88 CASE SUBSTR(instrng,curlet,1) = " "
89 CASE (colorvar)
90 @ 0,0 SAY " " COLOR (colorvar)
91 @ 0,3 SAY " " COLOR W+/N
92 @ 1,0 SAY " " COLOR W+/N
93 @ 1,3 SAY " " COLOR W+/N
94 @ 2,1 SAY " " COLOR W+/N
95 @ 3,0 SAY " " COLOR W+/N
96 @ 3,3 SAY " " COLOR W+/N
97
98 CASE SUBSTR(instrng,curlet,1) = "s"
99 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) :
100 FROM startx,curentry TO startx + 5,curentry + 5 ;
101 NONE NOSHADOW
102 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
103 curentry = curentry + 7
104 CASE SUBSTR(instrng,curlet,1) = " "
105 CASE (colorvar)
106 @ 1,0 SAY " " COLOR W+/N
107 @ 1,2 SAY " " COLOR W+/N
108 @ 1,3 SAY " " COLOR W+/N
109 @ 2,0 SAY " " COLOR W+/N
110 @ 2,2 SAY " " COLOR W+/N
111 @ 2,5 SAY " " COLOR W+/N
112 @ 3,0 SAY " " COLOR W+/N
113 @ 3,4 SAY " " COLOR W+/N
114 @ 4,0 SAY " " COLOR (colorvar)
115 CASE SUBSTR(instrng,curlet,1) = "e"
116 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) :
117 FROM startx,curentry TO startx + 5,curentry + 5 ;
118 NONE NOSHADOW
119 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
120 curentry = curentry + 7
121 CASE SUBSTR(instrng,curlet,1) = " "
122 CASE (colorvar)
123 @ 1,0 SAY " " COLOR W+/N
124 @ 1,2 SAY " " COLOR W+/N
125 @ 1,4 SAY " " COLOR W+/N
126 @ 2,0 SAY " " COLOR W+/N
127
128
129
130

```

```

131 @ 3,0 SAY " " COLOR W+/N
132 @ 3,2 SAY " " COLOR W+/N
133 @ 3,5 SAY " " COLOR (colorvar)
134
135 @ 4,0 SAY " " COLOR (colorvar)
136
137 -CASE SUBSTR(instring,curlet,1) = "n"
138 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
139 FROM startx,curlet TO startx + 5,curlet + 5 ;
140 NONE NOSHADOW
141 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
142 curlet = curlet + 6
143 @ 1,0 SAY " " COLOR W+/N
144 @ 1,3 SAY " " COLOR (colorvar)
145 @ 1,4 SAY " " COLOR (colorvar)
146
147 @ 2,0 SAY " " COLOR W+/N
148 @ 2,3 SAY " " COLOR (colorvar)
149
150 @ 3,0 SAY " " COLOR W+/N
151
152 @ 4,0 SAY " " COLOR (colorvar)
153
154 -CASE SUBSTR(instring,curlet,1) = "o"
155 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
156 FROM startx,curlet TO startx + 5,curlet + 5 ;
157 NONE NOSHADOW
158 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
159 curlet = curlet + 7
160 @ 1,0 SAY " "
161 @ 1,2 SAY " " COLOR W+/N
162 @ 1,4 SAY " "
163
164 @ 2,0 SAY " " " COLOR W+/N
165
166 @ 3,0 SAY " "
167 @ 3,2 SAY " " " COLOR W+/N
168
169 @ 4,0 SAY " " " COLOR (colorvar)
170
171 -CASE SUBSTR(instring,curlet,1) = "a"
172 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
173 FROM startx,curlet TO startx + 5,curlet + 6 ;
174 NONE NOSHADOW
175 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
176 curlet = curlet + 8
177 @ 1,0 SAY " "
178 @ 1,2 SAY " " " COLOR W+/N
179 @ 1,4 SAY " "
180
181 @ 2,0 SAY " " " " COLOR W+/N
182 @ 2,2 SAY " " " " COLOR W+/N
183
184 @ 3,0 SAY " "
185 @ 3,2 SAY " " " COLOR W+/N
186
187 @ 4,0 SAY " " " " COLOR (colorvar)
188
189 -CASE SUBSTR(instring,curlet,1) = "i"
190 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
191 FROM startx,curlet TO startx + 5,curlet + 5 ;
192 NONE NOSHADOW
193 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
194 curlet = curlet + 4
195 @ 0,1 SAY " " " COLOR W+/N
196 @ 0,2 SAY " " " COLOR (colorvar)
197
198 @ 1,0 SAY " "
199 @ 1,2 SAY " " " COLOR (colorvar)

```

```

197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262

```

```

@ 2,0 SAY " " " COLOR W+/N
@ 3,0 SAY " " " COLOR W+/N
@ 4,0 SAY " " " COLOR (colorvar)
CASE SUBSTR(instring,curlet,1) = "n"
DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
FROM startx,curlet TO startx + 5,curlet + 5 ;
NONE NOSHADOW
ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
curlet = curlet + 7
@ 1,0 SAY " "
@ 1,2 SAY " " " COLOR W+/N
@ 1,4 SAY " "
@ 2,0 SAY " " " " COLOR W+/N
@ 3,0 SAY " " " " COLOR W+/N
@ 4,0 SAY " " " " COLOR (colorvar)
CASE SUBSTR(instring,curlet,1) = "n"
DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
FROM startx,curlet TO startx + 5,curlet + 6 ;
NONE NOSHADOW
ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
curlet = curlet + 10
@ 1,0 SAY " "
@ 1,2 SAY " " " COLOR W+/N
@ 1,4 SAY " "
@ 1,5 SAY " " " COLOR W+/N
@ 1,7 SAY " "
@ 2,0 SAY " " " " COLOR W+/N
@ 3,0 SAY " " " " COLOR W+/N
@ 4,0 SAY " " " " COLOR (colorvar)
CASE SUBSTR(instring,curlet,1) = "n"
DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
FROM startx,curlet TO startx + 5,curlet + 6 ;
NONE NOSHADOW
ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
curlet = curlet + 7
@ 0,0 SAY " "
@ 0,3 SAY " " " COLOR W+/N
@ 0,5 SAY " " " COLOR (colorvar)
@ 1,0 SAY " "
@ 1,3 SAY " " " COLOR W+/N
@ 1,4 SAY " " " COLOR (colorvar)
@ 2,1 SAY " " " " COLOR W+/N
@ 3,1 SAY " " " " COLOR W+/N
@ 4,0 SAY " " " " COLOR (colorvar)
CASE SUBSTR(instring,curlet,1) = "i"
DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
FROM startx,curlet TO startx + 5,curlet + 5 ;
NONE NOSHADOW
ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
curlet = curlet + 4
@ 0,0 SAY " "
@ 0,2 SAY " " " COLOR (colorvar)

```

```

263 @ 1,0 SAY " " COLOR W+/N
264
265 @ 2,0 SAY " " COLOR W+/N
266
267 @ 3,0 SAY " " COLOR W+/N
268
269 @ 4,0 SAY " " COLOR (colorvar)
270
271 --CASE SUBSTR(instring,curlet,1) = "k"
272 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
273 FROM startx,curlet TO startx + 5,curlet + 7 ;
274 NONE NOSHADE
275 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
276 curlet = curlet + 8
277 @ 0,0 SAY " "
278 @ 0,2 SAY " " COLOR (colorvar)
279
280 @ 1,0 SAY " " COLOR W+/N
281 @ 1,5 SAY " " COLOR (colorvar)
282
283 @ 2,0 SAY " " COLOR W+/N
284 @ 2,4 SAY " " COLOR (colorvar)
285
286 @ 3,0 SAY " " COLOR W+/N
287 @ 3,3 SAY " "
288
289 @ 4,0 SAY " " COLOR (colorvar)
290
291 --CASE SUBSTR(instring,curlet,1) = "u"
292 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
293 FROM startx,curlet TO startx + 5,curlet + 6 ;
294 NONE NOSHADE
295 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
296 curlet = curlet + 7
297 @ 1,0 SAY " "
298 @ 1,2 SAY " " COLOR (colorvar)
299 @ 1,3 SAY " "
300 @ 1,5 SAY " " COLOR (colorvar)
301
302 @ 2,0 SAY " " COLOR W+/N
303
304 @ 3,0 SAY " " COLOR W+/N
305 @ 3,2 SAY " " COLOR W+/N
306 @ 3,3 SAY " "
307 @ 3,4 SAY " " COLOR W+/N
308
309 @ 4,0 SAY " " COLOR (colorvar)
310
311 --CASE SUBSTR(instring,curlet,1) = "b"
312 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
313 FROM startx,curlet TO startx + 5,curlet + 6 ;
314 NONE NOSHADE
315 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
316 curlet = curlet + 7
317 @ 0,0 SAY " "
318 @ 0,2 SAY " " COLOR (colorvar)
319
320 @ 1,0 SAY " " COLOR W+/N
321 @ 1,4 SAY " "
322
323 @ 2,0 SAY " " COLOR W+/N
324 @ 2,3 SAY " "
325 @ 2,5 SAY " " COLOR (colorvar)
326
327 @ 3,0 SAY " " COLOR W+/N
328
329 @ 4,0 SAY " " COLOR (colorvar)
330
331 --CASE SUBSTR(instring,curlet,1) = "n"
332 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
333 FROM startx,curlet TO startx + 5,curlet + 7 ;
334 NONE NOSHADE
335 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
336 curlet = curlet + 8
337 @ 0,0 SAY " "
338 @ 0,2 SAY " " COLOR (colorvar)
339
340 @ 1,0 SAY " " COLOR W+/N
341 @ 1,5 SAY " " COLOR (colorvar)
342
343 @ 2,0 SAY " " COLOR W+/N
344 @ 2,4 SAY " " COLOR (colorvar)
345
346 @ 3,0 SAY " " COLOR W+/N
347 @ 3,3 SAY " "
348
349 @ 4,0 SAY " " COLOR (colorvar)
350
351 --CASE SUBSTR(instring,curlet,1) = "q"
352 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
353 FROM startx,curlet TO startx + 5,curlet + 6 ;
354 NONE NOSHADE
355 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
356 curlet = curlet + 7
357 @ 1,0 SAY " "
358 @ 1,2 SAY " " COLOR (colorvar)
359 @ 1,3 SAY " "
360 @ 1,5 SAY " " COLOR (colorvar)
361
362 @ 2,0 SAY " " COLOR W+/N
363 @ 2,2 SAY " " COLOR W+/N
364 @ 2,3 SAY " "
365 @ 2,5 SAY " " COLOR W+/N
366
367 @ 3,0 SAY " " COLOR W+/N
368 @ 3,2 SAY " " COLOR W+/N
369 @ 3,3 SAY " "
370 @ 3,4 SAY " " COLOR W+/N
371
372 @ 4,0 SAY " " COLOR (colorvar)
373
374 --CASE SUBSTR(instring,curlet,1) = "g"
375 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
376 FROM startx,curlet TO startx + 5,curlet + 6 ;
377 NONE NOSHADE
378 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
379 curlet = curlet + 7
380 @ 0,0 SAY " "
381 @ 0,2 SAY " " COLOR (colorvar)
382
383 @ 1,0 SAY " " COLOR W+/N
384 @ 1,4 SAY " "
385
386 @ 2,0 SAY " " COLOR W+/N
387 @ 2,3 SAY " "
388 @ 2,5 SAY " " COLOR (colorvar)
389
390 @ 3,0 SAY " " COLOR W+/N
391
392 @ 4,0 SAY " " COLOR (colorvar)
393
394 --CASE SUBSTR(instring,curlet,1) = "c"
395 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;

```

```

329 FROM startx,curlet TO startx + 5,curlet + 5 ;
330 NONE NOSHADE
331 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
332 curlet = curlet + 6
333 @ 1,0 SAY " " COLOR W+/N
334 @ 1,2 SAY " " COLOR (colorvar)
335 @ 1,4 SAY " " COLOR (colorvar)
336
337 @ 2,0 SAY " " COLOR W+/N
338 @ 2,3 SAY " " COLOR (colorvar)
339
340 @ 3,0 SAY " " COLOR W+/N
341 @ 3,2 SAY " " COLOR (colorvar)
342 @ 3,4 SAY " " COLOR (colorvar)
343
344 @ 4,0 SAY " " COLOR (colorvar)
345
346 --CASE SUBSTR(instring,curlet,1) = "d"
347 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
348 FROM startx,curlet TO startx + 5,curlet + 5 ;
349 NONE NOSHADE
350 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
351 curlet = curlet + 7
352 @ 0,0 SAY " "
353 @ 0,5 SAY " " COLOR (colorvar)
354
355 @ 1,0 SAY " " COLOR W+/N
356 @ 1,2 SAY " " COLOR W+/N
357
358 @ 2,0 SAY " " COLOR W+/N
359 @ 2,2 SAY " " COLOR (colorvar)
360 @ 2,3 SAY " " COLOR W+/N
361
362 @ 3,0 SAY " " COLOR (colorvar)
363 @ 3,5 SAY " " COLOR (colorvar)
364
365 @ 4,0 SAY " " COLOR (colorvar)
366
367 --CASE SUBSTR(instring,curlet,1) = "m"
368 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
369 FROM startx,curlet TO startx + 5,curlet + 5 ;
370 NONE NOSHADE
371 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
372 curlet = curlet + 7
373 @ 1,0 SAY " " COLOR W+/N
374 @ 1,2 SAY " " COLOR W+/N
375 @ 1,3 SAY " "
376 @ 1,5 SAY " " COLOR (colorvar)
377
378 @ 2,0 SAY " " COLOR W+/N
379 @ 2,2 SAY " " COLOR (colorvar)
380 @ 2,3 SAY " " COLOR W+/N
381
382 @ 3,0 SAY " " COLOR (colorvar)
383 @ 3,5 SAY " " COLOR (colorvar)
384
385 @ 4,0 SAY " " COLOR (colorvar)
386 @ 4,3 SAY " " COLOR W+/N
387
388 @ 5,0 SAY " " COLOR (colorvar)
389
390 --CASE SUBSTR(instring,curlet,1) = "g"
391 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
392 FROM startx,curlet TO startx + 5,curlet + 6 ;
393 NONE NOSHADE
394 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
395 curlet = curlet + 7
396 @ 1,0 SAY " "

```

```

395 @ 1,2 SAY " " COLOR W+/N
396 @ 1,3 SAY " " COLOR (colorvar)
397 @ 1,5 SAY " " COLOR (colorvar)
398 @ 2,0 SAY " " COLOR W+/N
399 @ 2,2 SAY " " COLOR (colorvar)
400 @ 2,3 SAY " " COLOR W+/N
401 @ 3,0 SAY " " COLOR (colorvar)
402 @ 3,5 SAY " " COLOR (colorvar)
403 @ 4,0 SAY " " COLOR W+/N
404 @ 4,2 SAY " " COLOR W+/N
405 @ 4,3 SAY " " COLOR W+/N
406 @ 4,4 SAY " " COLOR W+/N
407 @ 5,0 SAY " " COLOR (colorvar)
408 -CASE SUBSTR(instr,curlet,1) = "h"
409 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
410 FROM startx,curlet + 5,curlet + 5 ;
411 NONE NOSHADE
412 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
413 currenty = curlet + 7
414 @ 0,0 SAY " " COLOR W+/N
415 @ 0,2 SAY " " COLOR (colorvar)
416 @ 1,0 SAY " " COLOR W+/N
417 @ 1,4 SAY " " COLOR W+/N
418 @ 2,0 SAY " " COLOR W+/N
419 @ 3,0 SAY " " COLOR W+/N
420 @ 4,0 SAY " " COLOR (colorvar)
421 -CASE SUBSTR(instr,curlet,1) = "j"
422 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
423 FROM startx,curlet + 5,curlet + 5 ;
424 NONE NOSHADE
425 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
426 currenty = curlet + 5
427 @ 0,0 SAY " " COLOR W+/N
428 @ 0,2 SAY " " COLOR (colorvar)
429 @ 0,3 SAY " " COLOR (colorvar)
430 @ 1,0 SAY " " COLOR (colorvar)
431 @ 1,3 SAY " " COLOR (colorvar)
432 @ 2,1 SAY " " COLOR W+/N
433 @ 3,1 SAY " " COLOR W+/N
434 @ 4,0 SAY " " COLOR W+/N
435 @ 4,1 SAY " " COLOR W+/N
436 @ 4,3 SAY " " COLOR (colorvar)
437 -CASE SUBSTR(instr,curlet,1) = "p"
438 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
439 FROM startx,curlet + 5,curlet + 6 ;
440 NONE NOSHADE
441 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
442 currenty = curlet + 7
443 @ 1,0 SAY " " COLOR W+/N
444 @ 2,0 SAY " " COLOR W+/N
445 @ 2,3 SAY " " COLOR W+/N
446 @ 2,5 SAY " " COLOR (colorvar)
447 @ 3,0 SAY " " COLOR W+/N
448 @ 3,3 SAY " " COLOR (colorvar)
449 @ 3,4 SAY " " COLOR W+/N
450 @ 3,6 SAY " " COLOR (colorvar)
451 @ 4,0 SAY " " COLOR (colorvar)
452 -CASE SUBSTR(instr,curlet,1) = "y"
453 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
454 FROM startx,curlet + 5,curlet + 6 ;
455 NONE NOSHADE
456 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
457 currenty = curlet + 7
458 @ 1,0 SAY " " COLOR W+/N
459 @ 2,0 SAY " " COLOR W+/N
460 @ 2,3 SAY " " COLOR W+/N
461 @ 2,5 SAY " " COLOR (colorvar)

```

```

461 @ 3,0 SAY " " COLOR W+/N
462 @ 4,0 SAY " " COLOR W+/N
463 @ 4,3 SAY " " COLOR (colorvar)
464 @ 5,0 SAY " " COLOR (colorvar)
465 -CASE SUBSTR(instr,curlet,1) = "w"
466 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
467 FROM startx,curlet + 5,curlet + 6 ;
468 NONE NOSHADE
469 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
470 currenty = curlet + 9
471 @ 1,0 SAY " " COLOR (colorvar)
472 @ 1,2 SAY " " COLOR (colorvar)
473 @ 1,3 SAY " " COLOR (colorvar)
474 @ 1,4 SAY " " COLOR (colorvar)
475 @ 1,5 SAY " " COLOR (colorvar)
476 @ 1,7 SAY " " COLOR (colorvar)
477 @ 2,0 SAY " " COLOR W+/N
478 @ 2,2 SAY " " COLOR W+/N
479 @ 3,1 SAY " " COLOR W+/N
480 @ 4,0 SAY " " COLOR (colorvar)
481 -CASE SUBSTR(instr,curlet,1) = "x"
482 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
483 FROM startx,curlet + 5,curlet + 7 ;
484 NONE NOSHADE
485 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
486 currenty = curlet + 8
487 @ 1,0 SAY " " COLOR W+/N
488 @ 1,5 SAY " " COLOR W+/N
489 @ 1,6 SAY " " COLOR (colorvar)
490 @ 2,0 SAY " " COLOR W+/N
491 @ 2,4 SAY " " COLOR W+/N
492 @ 2,5 SAY " " COLOR (colorvar)
493 @ 3,0 SAY " " COLOR W+/N
494 @ 3,3 SAY " " COLOR (colorvar)
495 @ 3,4 SAY " " COLOR W+/N
496 @ 3,6 SAY " " COLOR (colorvar)
497 @ 4,0 SAY " " COLOR (colorvar)
498 -CASE SUBSTR(instr,curlet,1) = "y"
499 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
500 FROM startx,curlet + 5,curlet + 6 ;
501 NONE NOSHADE
502 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
503 currenty = curlet + 7
504 @ 1,0 SAY " " COLOR (colorvar)
505 @ 1,2 SAY " " COLOR (colorvar)
506 @ 1,3 SAY " " COLOR (colorvar)
507 @ 1,5 SAY " " COLOR (colorvar)
508 @ 2,0 SAY " " COLOR W+/N
509 @ 3,0 SAY " " COLOR W+/N
510 @ 3,4 SAY " " COLOR W+/N
511 @ 4,0 SAY " " COLOR W+/N
512 @ 4,3 SAY " " COLOR W+/N
513 @ 5,0 SAY " " COLOR (colorvar)
514 -CASE SUBSTR(instr,curlet,1) = "y"
515 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
516 FROM startx,curlet + 5,curlet + 6 ;
517 NONE NOSHADE
518 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
519 currenty = curlet + 7
520 @ 1,0 SAY " " COLOR W+/N
521 @ 2,0 SAY " " COLOR W+/N
522 @ 2,3 SAY " " COLOR W+/N
523 @ 2,5 SAY " " COLOR (colorvar)
524 @ 3,0 SAY " " COLOR (colorvar)
525 -CASE SUBSTR(instr,curlet,1) = "y"
526

```

```

527 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
528 FROM startx,curynto TO startx + 5,curynto + 6 ;
529 NONE NOSHADOW
530 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
531 curynty = curynty + 7
532 @ 1,0 SAY " " COLOR (colorvar)
533 @ 1,2 SAY " " COLOR (colorvar)
534 @ 1,3 SAY " " COLOR (colorvar)
535 @ 1,5 SAY " " COLOR (colorvar)
536
537 @ 2,0 SAY " " COLOR (colorvar)
538 @ 2,1 SAY " " COLOR W+/N
539
540 @ 3,0 SAY " "
541 @ 3,3 SAY " " COLOR W+/N
542
543 @ 4,0 SAY " " COLOR (colorvar)
544
545 --CASE SUBSTR(instrng,curlet,1) = "n"
546 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
547 FROM startx,curynto TO startx + 5,curynto + 4 ;
548 NONE NOSHADOW
549 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
550 curynty = curynty + 6
551 @ 0,0 SAY " "
552 @ 0,4 SAY " " COLOR (colorvar)
553
554 @ 1,0 SAY " "
555 @ 1,3 SAY " " COLOR (colorvar)
556
557 @ 2,0 SAY " "
558 @ 2,2 SAY " " COLOR (colorvar)
559
560 @ 3,0 SAY " "
561 @ 3,1 SAY " " COLOR (colorvar)
562
563 @ 4,0 SAY " " COLOR (colorvar)
564
565 --CASE SUBSTR(instrng,curlet,1) = "z"
566 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
567 FROM startx,curynto TO startx + 5,curynto + 6 ;
568 NONE NOSHADOW
569 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
570 curynty = curynty + 7
571 @ 1,0 SAY " "
572 @ 1,1 SAY " " COLOR W+/N
573 @ 1,5 SAY " " COLOR (colorvar)
574
575 @ 2,0 SAY " "
576 @ 2,3 SAY " " COLOR W+/N
577 @ 2,5 SAY " " COLOR (colorvar)
578
579 @ 3,0 SAY " "
580 @ 3,4 SAY " " COLOR W+/N
581
582 @ 4,0 SAY " " COLOR (colorvar)
583 startx = startx + 6
584 curynty = startx
585 --CASE SUBSTR(instrng,curlet,1) = "A"
586 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
587 FROM startx,curynto TO startx + 5,curynto + 7 ;
588 NONE NOSHADOW
589 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
590 curynty = curynty + 9
591 @ 0,2 SAY " " COLOR W+/N
592 @ 0,5 SAY " " COLOR (colorvar)

```

```

593
594 @ 1,1 SAY " " COLOR W+/N
595 @ 1,6 SAY " " COLOR (colorvar)
596
597 @ 2,0 SAY " "
598 @ 2,1 SAY " " COLOR W+/N
599
600 @ 3,0 SAY " " COLOR W+/N
601 @ 3,3 SAY " " COLOR (colorvar)
602 @ 3,5 SAY " " COLOR W+/N
603
604 @ 4,0 SAY " " COLOR (colorvar)
605
606 --CASE SUBSTR(instrng,curlet,1) = "B"
607 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
608 FROM startx,curynto TO startx + 5,curynto + 5 ;
609 NONE NOSHADOW
610 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
611 curynty = curynty + 8
612 @ 0,0 SAY " " COLOR W+/N
613 @ 0,4 SAY " "
614
615 @ 1,0 SAY " " COLOR W+/N
616 @ 1,3 SAY " "
617 @ 1,5 SAY " " COLOR W+/N
618
619 @ 2,0 SAY " " COLOR W+/N
620 @ 2,3 SAY " " COLOR (colorvar)
621 @ 2,4 SAY " " COLOR W+/N
622 @ 2,5 SAY " " COLOR (colorvar)
623
624 @ 3,0 SAY " " COLOR W+/N
625 @ 3,3 SAY " " COLOR W+/N
626
627 @ 4,0 SAY " " COLOR (colorvar)
628
629 --CASE SUBSTR(instrng,curlet,1) = "C"
630 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
631 FROM startx,curynto TO startx + 5,curynto + 8 ;
632 NONE NOSHADOW
633 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
634 curynty = curynty + 9
635 @ 0,0 SAY " "
636 @ 0,3 SAY " " COLOR W+/N
637 @ 0,6 SAY " "
638
639 @ 1,0 SAY " " COLOR W+/N
640 @ 1,3 SAY " " COLOR (colorvar)
641 @ 1,4 SAY " "
642 @ 1,7 SAY " " COLOR (colorvar)
643
644 @ 2,0 SAY " " COLOR W+/N
645 @ 2,3 SAY " "
646
647 @ 3,0 SAY " "
648 @ 3,6 SAY " " COLOR W+/N
649
650 @ 4,0 SAY " " COLOR (colorvar)
651
652 --CASE SUBSTR(instrng,curlet,1) = "D"
653 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
654 FROM startx,curynto TO startx + 5,curynto + 6 ;
655 NONE NOSHADOW
656 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
657 curynty = curynty + 8
658 @ 0,0 SAY " " COLOR W+/N
659 @ 0,4 SAY " "
660
661 @ 1,0 SAY " " COLOR W+/N

```

```

659 @ 1,3 SAY " " COLOR (colorvar)
660 @ 1,6 SAY " " COLOR (colorvar)
661
662 @ 2,0 SAY " " COLOR W+N
663 @ 2,3 SAY " "
664 @ 2,6 SAY " " COLOR (colorvar)
665
666 @ 3,0 SAY " " COLOR W+N
667 @ 3,6 SAY " " COLOR (colorvar)
668
669 @ 4,0 SAY " " COLOR (colorvar)
670
671 -CASE SUBSTR(instr,curlet,1) = "E"
672 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
673 FROM startx,curynto TO startx + 5,curynto + 6 ;
674 NONE NOSHADOW
675 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
676 curlynto = curlynto + 7
677 @ 0,0 SAY " " COLOR W+N
678 @ 0,5 SAY " " COLOR (colorvar)
679
680 @ 1,0 SAY " " COLOR W+N
681 @ 1,3 SAY " "
682
683 @ 2,0 SAY " " COLOR W+N
684 @ 2,3 SAY " " COLOR (colorvar)
685
686 @ 3,0 SAY " " COLOR W+N
687 @ 3,3 SAY " "
688
689 @ 4,0 SAY " " COLOR (colorvar)
690
691 -CASE SUBSTR(instr,curlet,1) = "F"
692 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
693 FROM startx,curynto TO startx + 5,curynto + 6 ;
694 NONE NOSHADOW
695 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
696 curlynto = curlynto + 7
697 @ 0,0 SAY " " COLOR W+N
698 @ 0,5 SAY " " COLOR (colorvar)
699
700 @ 1,0 SAY " " COLOR W+N
701 @ 1,3 SAY " "
702
703 @ 2,0 SAY " " COLOR W+N
704 @ 2,3 SAY " " COLOR (colorvar)
705
706 @ 3,0 SAY " " COLOR W+N
707
708 @ 4,0 SAY " " COLOR (colorvar)
709
710 -CASE SUBSTR(instr,curlet,1) = "G"
711 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
712 FROM startx,curynto TO startx + 5,curynto + 8 ;
713 NONE NOSHADOW
714 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
715 curlynto = curlynto + 9
716 @ 0,0 SAY " "
717 @ 0,3 SAY " " COLOR W+N
718 @ 0,6 SAY " "
719
720 @ 1,0 SAY " " COLOR W+N
721 @ 1,3 SAY " " COLOR (colorvar)
722 @ 1,4 SAY " "
723 @ 1,7 SAY " " COLOR (colorvar)
724
725 @ 2,0 SAY " " COLOR W+N
726 @ 2,3 SAY " "
727 @ 2,7 SAY " " COLOR (colorvar)

```

```

728
729 @ 3,0 SAY " " COLOR W+N
730 @ 3,6 SAY " " COLOR (colorvar)
731
732 @ 4,0 SAY " " COLOR (colorvar)
733
734 -CASE SUBSTR(instr,curlet,1) = "H"
735 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
736 FROM startx,curynto TO startx + 5,curynto + 7 ;
737 NONE NOSHADOW
738 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
739 curlynto = curlynto + 8
740 @ 0,0 SAY " "
741 @ 0,2 SAY " " COLOR (colorvar)
742 @ 0,4 SAY " "
743 @ 0,6 SAY " " COLOR (colorvar)
744
745 @ 1,0 SAY " " COLOR W+N
746 @ 1,3 SAY " "
747 @ 1,6 SAY " " COLOR (colorvar)
748
749 @ 2,0 SAY " " COLOR W+N
750 @ 2,3 SAY " " COLOR (colorvar)
751 @ 2,4 SAY " " COLOR W+N
752
753 @ 3,0 SAY " " COLOR W+N
754 @ 3,3 SAY " "
755 @ 3,4 SAY " " COLOR W+N
756
757 @ 4,0 SAY " " COLOR (colorvar)
758
759 -CASE SUBSTR(instr,curlet,1) = "I"
760 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
761 FROM startx,curynto TO startx + 5,curynto + 5 ;
762 NONE NOSHADOW
763 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
764 curlynto = curlynto + 4
765 @ 0,0 SAY " "
766 @ 0,2 SAY " " COLOR (colorvar)
767
768 @ 1,0 SAY " " COLOR W+N
769 @ 2,0 SAY " " COLOR W+N
770 @ 3,0 SAY " " COLOR W+N
771
772 @ 4,0 SAY " " COLOR (colorvar)
773
774 -CASE SUBSTR(instr,curlet,1) = "J"
775 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
776 FROM startx,curynto TO startx + 5,curynto + 6 ;
777 NONE NOSHADOW
778 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
779 curlynto = curlynto + 7
780 @ 0,0 SAY " "
781 @ 0,5 SAY " " COLOR (colorvar)
782
783 @ 1,3 SAY " " COLOR W+N
784 @ 2,3 SAY " " COLOR W+N
785
786 @ 3,0 SAY " "
787 @ 3,4 SAY " " COLOR W+N
788
789 @ 4,0 SAY " " COLOR (colorvar)
790
791 -CASE SUBSTR(instr,curlet,1) = "K"
792 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
793 FROM startx,curynto TO startx + 5,curynto + 7 ;
794 NONE NOSHADOW

```

```

791 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
792   curentry = curentry + 9
793   @ 0,0 SAY " " COLOR (colorvar)
794   @ 0,2 SAY " " COLOR (colorvar)
795   @ 0,3 SAY " " COLOR (colorvar)
796   @ 0,5 SAY " " COLOR (colorvar)
797   @ 0,6 SAY " " COLOR (colorvar)
798   @ 1,0 SAY " " COLOR W+/N
799   @ 1,5 SAY " " COLOR (colorvar)
800   @ 2,0 SAY " " COLOR W+/N
801   @ 2,4 SAY " " COLOR W+/N
802   @ 3,0 SAY " " COLOR W+/N
803   @ 3,3 SAY " "
804   @ 4,0 SAY " " COLOR (colorvar)
805   @ 4,2 SAY " "
806   @ 4,3 SAY " "
807   @ 4,5 SAY " "
808   @ 4,6 SAY " "
809   @ 4,7 SAY " "
810   @ 4,8 SAY " "
811   @ 4,9 SAY " "
812   @ 5,0 SAY " "
813   @ 5,2 SAY " "
814   @ 5,3 SAY " "
815   @ 5,5 SAY " "
816   @ 5,6 SAY " "
817   @ 5,7 SAY " "
818   @ 5,8 SAY " "
819   @ 5,9 SAY " "
820   @ 6,0 SAY " "
821   @ 6,2 SAY " "
822   @ 6,3 SAY " "
823   @ 6,5 SAY " "
824   @ 6,6 SAY " "
825   @ 6,7 SAY " "
826   @ 6,8 SAY " "
827   @ 6,9 SAY " "
828   @ 7,0 SAY " "
829   @ 7,2 SAY " "
830   @ 7,3 SAY " "
831   @ 7,5 SAY " "
832   @ 7,6 SAY " "
833   @ 7,7 SAY " "
834   @ 7,8 SAY " "
835   @ 7,9 SAY " "
836   @ 8,0 SAY " "
837   @ 8,2 SAY " "
838   @ 8,3 SAY " "
839   @ 8,5 SAY " "
840   @ 8,6 SAY " "
841   @ 8,7 SAY " "
842   @ 8,8 SAY " "
843   @ 8,9 SAY " "
844   @ 9,0 SAY " "
845   @ 9,2 SAY " "
846   @ 9,3 SAY " "
847   @ 9,5 SAY " "
848   @ 9,6 SAY " "
849   @ 9,7 SAY " "
850   @ 9,8 SAY " "
851   @ 9,9 SAY " "
852   @ 10,0 SAY " "
853   @ 10,2 SAY " "
854   @ 10,3 SAY " "
855   @ 10,5 SAY " "
856   @ 10,6 SAY " "

```

```

857   @ 1,0 SAY " " COLOR W+/N
858   @ 1,4 SAY " " COLOR (colorvar)
859   @ 1,7 SAY " " COLOR (colorvar)
860   @ 2,0 SAY " " COLOR W+/N
861   @ 2,3 SAY " "
862   @ 2,4 SAY " " COLOR W+/N
863   @ 2,6 SAY " "
864   @ 3,0 SAY " " COLOR W+/N
865   @ 3,3 SAY " "
866   @ 3,7 SAY " " COLOR W+/N
867   @ 4,0 SAY " " COLOR (colorvar)
868   @ 4,2 SAY " "
869   @ 4,3 SAY " "
870   @ 4,5 SAY " "
871   @ 4,6 SAY " "
872   @ 4,7 SAY " "
873   @ 4,8 SAY " "
874   @ 4,9 SAY " "
875   @ 5,0 SAY " "
876   @ 5,2 SAY " "
877   @ 5,3 SAY " "
878   @ 5,5 SAY " "
879   @ 5,6 SAY " "
880   @ 5,7 SAY " "
881   @ 5,8 SAY " "
882   @ 5,9 SAY " "
883   @ 6,0 SAY " "
884   @ 6,2 SAY " "
885   @ 6,3 SAY " "
886   @ 6,5 SAY " "
887   @ 6,6 SAY " "
888   @ 6,7 SAY " "
889   @ 6,8 SAY " "
890   @ 6,9 SAY " "
891   @ 7,0 SAY " "
892   @ 7,2 SAY " "
893   @ 7,3 SAY " "
894   @ 7,5 SAY " "
895   @ 7,6 SAY " "
896   @ 7,7 SAY " "
897   @ 7,8 SAY " "
898   @ 7,9 SAY " "
899   @ 8,0 SAY " "
900   @ 8,2 SAY " "
901   @ 8,3 SAY " "
902   @ 8,5 SAY " "
903   @ 8,6 SAY " "
904   @ 8,7 SAY " "
905   @ 8,8 SAY " "
906   @ 8,9 SAY " "
907   @ 9,0 SAY " "
908   @ 9,2 SAY " "
909   @ 9,3 SAY " "
910   @ 9,5 SAY " "
911   @ 9,6 SAY " "
912   @ 9,7 SAY " "
913   @ 9,8 SAY " "
914   @ 9,9 SAY " "
915   @ 10,0 SAY " "
916   @ 10,2 SAY " "
917   @ 10,3 SAY " "
918   @ 10,5 SAY " "
919   @ 10,6 SAY " "
920   @ 10,7 SAY " "
921   @ 10,8 SAY " "
922   @ 10,9 SAY " "

```



```

923 @ 1,0 SAY " " COLOR W+/N
924 @ 1,3 SAY " " COLOR (colorvar)
925 @ 1,5 SAY " " COLOR (colorvar)
926 @ 1,7 SAY " " COLOR (colorvar)
927 @ 1,9 SAY " " COLOR (colorvar)
928 @ 2,0 SAY " " COLOR W+/N
929 @ 2,3 SAY " " COLOR (colorvar)
930 @ 2,6 SAY " " COLOR (colorvar)
931 @ 2,9 SAY " " COLOR (colorvar)
932 @ 3,0 SAY " " COLOR (colorvar)
933 @ 3,3 SAY " " COLOR (colorvar)
934 @ 3,6 SAY " " COLOR (colorvar)
935 @ 4,0 SAY " " COLOR (colorvar)
936 @ 4,3 SAY " " COLOR (colorvar)
937 @ 4,6 SAY " " COLOR (colorvar)
938 @ 4,9 SAY " " COLOR (colorvar)
939 @ 5,0 SAY " " COLOR (colorvar)
940 @ 5,3 SAY " " COLOR (colorvar)
941 @ 5,6 SAY " " COLOR (colorvar)
942 @ 6,0 SAY " " COLOR (colorvar)
943 @ 6,3 SAY " " COLOR (colorvar)
944 @ 6,6 SAY " " COLOR (colorvar)
945 @ 6,9 SAY " " COLOR (colorvar)
946 @ 7,0 SAY " " COLOR (colorvar)
947 @ 7,3 SAY " " COLOR (colorvar)
948 @ 7,6 SAY " " COLOR (colorvar)
949 @ 7,9 SAY " " COLOR (colorvar)
950 @ 8,0 SAY " " COLOR (colorvar)
951 @ 8,3 SAY " " COLOR (colorvar)
952 @ 8,6 SAY " " COLOR (colorvar)
953 @ 8,9 SAY " " COLOR (colorvar)
954 @ 9,0 SAY " " COLOR (colorvar)
955 @ 9,3 SAY " " COLOR (colorvar)
956 @ 9,6 SAY " " COLOR (colorvar)
957 @ 9,9 SAY " " COLOR (colorvar)
958 @ 10,0 SAY " " COLOR (colorvar)
959 @ 10,3 SAY " " COLOR (colorvar)
960 @ 10,6 SAY " " COLOR (colorvar)
961 @ 10,9 SAY " " COLOR (colorvar)
962 @ 11,0 SAY " " COLOR (colorvar)
963 @ 11,3 SAY " " COLOR (colorvar)
964 @ 11,6 SAY " " COLOR (colorvar)
965 @ 11,9 SAY " " COLOR (colorvar)
966 @ 12,0 SAY " " COLOR (colorvar)
967 @ 12,3 SAY " " COLOR (colorvar)
968 @ 12,6 SAY " " COLOR (colorvar)
969 @ 12,9 SAY " " COLOR (colorvar)
970 @ 13,0 SAY " " COLOR (colorvar)
971 @ 13,3 SAY " " COLOR (colorvar)
972 @ 13,6 SAY " " COLOR (colorvar)
973 @ 13,9 SAY " " COLOR (colorvar)
974 @ 14,0 SAY " " COLOR (colorvar)
975 @ 14,3 SAY " " COLOR (colorvar)
976 @ 14,6 SAY " " COLOR (colorvar)
977 @ 14,9 SAY " " COLOR (colorvar)
978 @ 15,0 SAY " " COLOR (colorvar)
979 @ 15,3 SAY " " COLOR (colorvar)
980 @ 15,6 SAY " " COLOR (colorvar)
981 @ 15,9 SAY " " COLOR (colorvar)
982 @ 16,0 SAY " " COLOR (colorvar)
983 @ 16,3 SAY " " COLOR (colorvar)
984 @ 16,6 SAY " " COLOR (colorvar)
985 @ 16,9 SAY " " COLOR (colorvar)
986 @ 17,0 SAY " " COLOR (colorvar)
987 @ 17,3 SAY " " COLOR (colorvar)
988 @ 17,6 SAY " " COLOR (colorvar)

```

```

989 @ 1,2 SAY " " COLOR W+/N
990 @ 1,5 SAY " " COLOR W+/N
991 @ 1,8 SAY " " COLOR W+/N
992 @ 2,1 SAY " " COLOR W+/N
993 @ 2,4 SAY " " COLOR W+/N
994 @ 2,7 SAY " " COLOR W+/N
995 @ 3,0 SAY " " COLOR (colorvar)
996 @ 3,3 SAY " " COLOR (colorvar)
997 @ 3,6 SAY " " COLOR (colorvar)
998 @ 3,9 SAY " " COLOR (colorvar)
999 @ 4,2 SAY " " COLOR (colorvar)
1000 @ 4,5 SAY " " COLOR (colorvar)
1001 @ 4,8 SAY " " COLOR (colorvar)
1002 @ 5,1 SAY " " COLOR (colorvar)
1003 @ 5,4 SAY " " COLOR (colorvar)
1004 @ 5,7 SAY " " COLOR (colorvar)
1005 @ 6,0 SAY " " COLOR (colorvar)
1006 @ 6,3 SAY " " COLOR (colorvar)
1007 @ 6,6 SAY " " COLOR (colorvar)
1008 @ 6,9 SAY " " COLOR (colorvar)
1009 @ 7,2 SAY " " COLOR (colorvar)
1010 @ 7,5 SAY " " COLOR (colorvar)
1011 @ 7,8 SAY " " COLOR (colorvar)
1012 @ 8,1 SAY " " COLOR (colorvar)
1013 @ 8,4 SAY " " COLOR (colorvar)
1014 @ 8,7 SAY " " COLOR (colorvar)
1015 @ 9,0 SAY " " COLOR (colorvar)
1016 @ 9,3 SAY " " COLOR (colorvar)
1017 @ 9,6 SAY " " COLOR (colorvar)
1018 @ 9,9 SAY " " COLOR (colorvar)
1019 @ 10,2 SAY " " COLOR (colorvar)
1020 @ 10,5 SAY " " COLOR (colorvar)
1021 @ 10,8 SAY " " COLOR (colorvar)
1022 @ 11,1 SAY " " COLOR (colorvar)
1023 @ 11,4 SAY " " COLOR (colorvar)
1024 @ 11,7 SAY " " COLOR (colorvar)
1025 @ 12,0 SAY " " COLOR (colorvar)
1026 @ 12,3 SAY " " COLOR (colorvar)
1027 @ 12,6 SAY " " COLOR (colorvar)
1028 @ 12,9 SAY " " COLOR (colorvar)
1029 @ 13,2 SAY " " COLOR (colorvar)
1030 @ 13,5 SAY " " COLOR (colorvar)
1031 @ 13,8 SAY " " COLOR (colorvar)
1032 @ 14,1 SAY " " COLOR (colorvar)
1033 @ 14,4 SAY " " COLOR (colorvar)
1034 @ 14,7 SAY " " COLOR (colorvar)
1035 @ 15,0 SAY " " COLOR (colorvar)
1036 @ 15,3 SAY " " COLOR (colorvar)
1037 @ 15,6 SAY " " COLOR (colorvar)
1038 @ 15,9 SAY " " COLOR (colorvar)
1039 @ 16,2 SAY " " COLOR (colorvar)
1040 @ 16,5 SAY " " COLOR (colorvar)
1041 @ 16,8 SAY " " COLOR (colorvar)
1042 @ 17,1 SAY " " COLOR (colorvar)
1043 @ 17,4 SAY " " COLOR (colorvar)
1044 @ 17,7 SAY " " COLOR (colorvar)
1045 @ 18,0 SAY " " COLOR (colorvar)
1046 @ 18,3 SAY " " COLOR (colorvar)
1047 @ 18,6 SAY " " COLOR (colorvar)
1048 @ 18,9 SAY " " COLOR (colorvar)
1049 @ 19,2 SAY " " COLOR (colorvar)
1050 @ 19,5 SAY " " COLOR (colorvar)
1051 @ 19,8 SAY " " COLOR (colorvar)
1052 @ 20,1 SAY " " COLOR (colorvar)
1053 @ 20,4 SAY " " COLOR (colorvar)
1054 @ 20,7 SAY " " COLOR (colorvar)

```

```

1055 @ 0,8 SAY " " COLOR (colorvar)
1056 @ 0,10 SAY " " COLOR (colorvar)
1057
1058 @ 1,0 SAY " " COLOR W+/N
1059 @ 1,2 SAY " " COLOR W+/N
1060 @ 1,3 SAY " " COLOR (colorvar)
1061 @ 1,6 SAY " " COLOR (colorvar)
1062 @ 1,7 SAY " " COLOR W+/N
1063 @ 1,9 SAY " " COLOR W+/N
1064
1065 @ 2,1 SAY " " COLOR W+/N
1066 @ 3,2 SAY " " COLOR W+/N
1067 @ 3,6 SAY " " COLOR W+/N
1068 @ 3,9 SAY " " COLOR (colorvar)
1069
1070 @ 4,0 SAY " " COLOR (colorvar)
1071
1072 -CASE SUBSTR(instring,curlet,1) = "X"
1073   DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
1074   FROM startx, curenty TO startx + 5, curenty + 8 ;
1075   NONE NOSHADOW
1076   ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
1077   curenty = curenty + 9
1078   @ 0,0 SAY " " COLOR (colorvar)
1079   @ 0,2 SAY " " COLOR (colorvar)
1080   @ 0,3 SAY " " COLOR (colorvar)
1081   @ 0,7 SAY " " COLOR (colorvar)
1082
1083   @ 1,0 SAY " "
1084   @ 1,5 SAY " " COLOR W+/N
1085   @ 1,6 SAY " " COLOR (colorvar)
1086
1087   @ 2,0 SAY " "
1088   @ 2,3 SAY " " COLOR W+/N
1089
1090   @ 3,0 SAY " " COLOR W+/N
1091   @ 3,3 SAY " " COLOR (colorvar)
1092   @ 3,4 SAY " "
1093   @ 3,7 SAY " " COLOR (colorvar)
1094
1095   @ 4,0 SAY " " COLOR (colorvar)
1096
1097 -CASE SUBSTR(instring,curlet,1) = "Y"
1098   DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
1099   FROM startx, curenty TO startx + 5, curenty + 9 ;
1100   NONE NOSHADOW
1101   ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
1102   curenty = curenty + 10
1103   @ 0,0 SAY " "
1104   @ 0,2 SAY " " COLOR (colorvar)
1105   @ 0,3 SAY " "
1106   @ 0,8 SAY " " COLOR (colorvar)
1107
1108   @ 1,0 SAY " "
1109   @ 1,3 SAY " " COLOR (colorvar)
1110   @ 1,4 SAY " "
1111   @ 1,7 SAY " " COLOR (colorvar)
1112
1113   @ 2,0 SAY " "
1114   @ 2,5 SAY " " COLOR W+/N
1115   @ 2,7 SAY " " COLOR (colorvar)
1116
1117   @ 3,3 SAY " " COLOR W+/N
1118
1119   @ 4,0 SAY " " COLOR (colorvar)
1120 -CASE SUBSTR(instring,curlet,1) = "Z"
1121   DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;

```

```

1121 FROM startx, curenty TO startx + 5, curenty + 6 ;
1122 NONE NOSHADOW
1123 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
1124 curenty = curenty + 8
1125 @ 0,0 SAY " " COLOR W+/N
1126 @ 0,1 SAY " " COLOR (colorvar)
1127 @ 0,6 SAY " " COLOR (colorvar)
1128
1129 @ 1,0 SAY " "
1130 @ 1,4 SAY " " COLOR W+/N
1131 @ 1,6 SAY " " COLOR (colorvar)
1132
1133 @ 2,0 SAY " "
1134 @ 2,3 SAY " " COLOR W+/N
1135
1136 @ 3,0 SAY " " COLOR W+/N
1137 @ 3,4 SAY " "
1138
1139 @ 4,0 SAY " " COLOR (colorvar)
1140 -ENDCASE
1141 curlet = curlet + 1
1142 -ENDDO
1143
1144 DEFINE WINDOW ("WRN" ) ;
1145 FROM x_coor+startx,y_coor-12 TO x_coor+startx + 11,y_coor + 60 ;
1146 NONE NOSHADOW
1147 ACTIVATE WINDOW ("WRN")
1148
1149 @ 2,17 SAY "Version 1.2 Date: Sept. 30, 1993"
1150 @ 3,17 SAY "(c) Naval Health Research Center."
1151
1152 @ 6,0 SAY "*****"
1153 => *****
1154 @ 7,0 SAY "UNAUTHORIZED ACCESS TO THIS UNITED STATES GOVERNMENT COMPU
1155 => TER SYSTEM"
1156 @ 8,0 SAY "AND/OR SOFTWARE IS PROHIBITED BY TITLE 18, UNITED STATES C
1157 => ODE, SECTION"
1158 @ 9,0 SAY "1030. THIS SYSTEM MAY ONLY BE USED FOR UNCLASSIFIED OFFIC
1159 => IAL BUSINESS."
1160 @ 10,0 SAY "*****"
1161 => *****
1162
1163 WAIT "" TIMEOUT 0 t
1164 IF NOT EMPTY(instring)
1165   curlet = 1
1166   FOR i = 1 TO len str
1167     DEACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
1168     RELEASE WINDOW ("letter" + ALLTRIM(STR(curlet)))
1169     curlet = curlet + 1
1170   NEXT
1171 ENDIF
1172 DEACTIVATE WINDOW "WRN"
1173 RELEASE WINDOW "WRN"
1174 CLEAR
1175 *: EOF: BIGCHARS.ACT

```

```

1  *
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 #REGION 1
27 PARAMETER arraylist, msg
28 EXTERNAL ARRAY arraylist
29 IF PARAMETER() < 2
30 RETURN ""
31 ENDIF
32
33 #REGION 0
34 REGIONAL m.currarea, m.talkstat, m.compstat
35
36 IF SET("TALK") = "ON"
37 SET TALK OFF
38 m.talkstat = "ON"
39 ELSE
40 m.talkstat = "OFF"
41 ENDIF
42 m.compstat = SET("COMPATIBLE")
43 SET COMPATIBLE FOXPLUS
44
45 m.currarea = SELECT()
46
47
48
49
50
51
52
53
54
55
56 IF NOT WEXIST("bselect")
57 DEFINE WINDOW bselect ;
58 FROM INT((SROW()-8)/2), INT((SCOL()-60)/2) ;
59 TO INT((SROW()-8)/2)+7, INT((SCOL()-60)/2)+59 ;
60 TITLE "Boot Strap" ;
61 NOFLOAT ;
62 NOCLOSE ;
63 SHADOW ;
64 DOUBLE ;
65 COLOR SCHEME 5
66 ENDIF

```

BSELECT.ACT 12-1-93 11:32a

```

67
68
69
70
71
72
73
74
75
76 #REGION 1
77 U = ""
78 m.text = ""
79
80
81
82
83
84
85
86
87
88 #REGION 1
89 IF WVISIBLE("bselect")
90 ACTIVATE WINDOW bselect SAME
91 ELSE
92 ACTIVATE WINDOW bselect NOSHOW
93 ENDIF
94 a 1,0 GET m.fl ;
95 PICTURE "g" ;
96 FROM arraylist ;
97 SIZE 3,57 ;
98 DEFAULT 1 ;
99 COLOR SCHEME 5, 6
100 a 5,21 GET m.action ;
101 PICTURE "a"HT \<Ok;\<Cancel" ;
102 SIZE 1,8,1 ;
103 DEFAULT 1 ;
104 VALID _q8q0r6sbm()
105 a 0,0 SAY m.msg ;
106 SIZE 1,37
107
108 IF NOT WVISIBLE("bselect")
109 ACTIVATE WINDOW bselect
110 ENDIF
111 READ CYCLE MODAL
112
113 RELEASE WINDOW bselect
114 SELECT (m.currarea)
115
116
117 #REGION 0
118 IF m.talkstat = "ON"
119 SET TALK ON
120 ENDIF
121 IF m.compstat = "ON"
122 SET COMPATIBLE ON
123 ENDIF
124
125
126
127
128
129
130
131
132

```

Page 1 of 2

BSELECT Setup Code - SECTION 2

BSELECT Screen Layout

BSELECT Cleanup Code

```

133 #REGION 1
134 RETURN m.text
135
136 *
137 *
138 *
139 *
140 *
141 *
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 FUNCTION _q8q0n6sbm  && m.action VALID
151 #REGION 1
152 IF m.action = 1
153   m.text = arraylist[m.f1,1]
154 ELSE
155   m.text = ""
156 ENDIF
157 *: EOF: BSELECT.ACT
158

```

_q8q0n6sbm	m.action VALID	
Function Origin:		
From Screen:	BSELECT,	Record Number: 3
Variable:	m.action	
Called By:	VALID Clause	
Object Type:	Push Button	
Snippet Number:	1	

```

65 | m.message = SUBSTR( m.message, 1,60 )
66 |-----|
67 |-----|
68 |-----|
69 |-----|
70 |-----|
71 |REGION 0
72 |REGIONAL m.curraarea, m.talkstat, m.compstat
73 |SET TALK OFF
74 |
75 |-----|
76 |-----|
77 |-----|
78 |-----|
79 |-----|
80 |-----|
81 |-----|
82 |-----|
83 |-----|
84 | *
85 | *
86 | *
87 | *
88 | *
89 | *
90 | *
91 | *
92 | *
93 | *
94 | *
95 | *
96 | *
97 | *
98 | *
99 | *
100 | *
101 | *
102 | *
103 | *
104 | *
105 | *
106 | *
107 | *
108 | *
109 | *
110 | *
111 | *
112 | *
113 | *
114 | *
115 | *
116 | *
117 | *
118 | *
119 | *
120 | *
121 | *
122 | *
123 | *
124 | *
125 | *
126 | *
127 | *
128 | *
129 | *
130 | *

```


257 *
258 * When Code from screen: CHOOSE
259 *
260 #REGION 1
261 * Start the READ on the list box
262 CURSOR = 08JNUM(m.choice)
263 *; EOF: CHOOSE.ACT

09/02/92	BUFACT.PRG	11:04:28
Author's Name		
Copyright (c) 1992 Company Name		
Address		
City,	Zip	
Description: This program was automatically generated by GENSCRN.		

BUFACT Setup Code - SECTION 1

```

1  *
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 #REGION 1
27 PARAMETER array1, array2
28 EXTERNAL ARRAY array1, array2
29 IF PARAMETER() < 2
30 RETURN ""
31 ENDIF
32
33 #REGION 0
34 REGIONAL m.curarea, m.talkstat, m.compstat
35
36 IF SET("TALK") = "ON"
37 SET TALK OFF
38 m.talkstat = "ON"
39 ELSE
40 m.talkstat = "OFF"
41 ENDIF
42 m.compstat = SET("COMPATIBLE")
43 SET COMPATIBLE FOXPLUS
44
45 m.curarea = SELECT()
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

Window definitions

```

67 IF NOT EXISTS("bfact")
68 DEFINE WINDOW bfact
69 FROM INT((SCOL()-1)/2), INT((SCOL()-64)/2) ;
70 TO INT((SCOL()-11)/2)+10, INT((SCOL()-64)/2)+63 ;
71 TITLE "Boot strap" ;
72 NOFLOAT ;
73 NOCLOSE ;
74 SHADOW ;
75 DOUBLE ;
76 COLOR SCHEME 5
77 ENDIF

```

BUFACT Setup Code - SECTION 2

BUFACT Screen Layout

```

67 *
68 *
69 *
70 *
71 *
72 *
73 *
74 *
75 #REGION 1
76 U = ""
77 m.text = ""
78
79 *
80 *
81 *
82 *
83 *
84 *
85 *
86 *
87 *
88 *
89 #REGION 1
90 IF NOT EXISTS("bfact")
91 ACTIVATE WINDOW bfact SAME
92 ELSE
93 ACTIVATE WINDOW bfact NOSHOW
94 ENDIF
95 @ 1,18 GET m.f1 ;
96 PICTURE "g" ;
97 FROM array1 ;
98 SIZE 3,41 ;
99 DEFAULT 1 ;
100 COLOR SCHEME 5, 6
101 @ 4,18 GET m.f2 ;
102 PICTURE "g" ;
103 FROM array2 ;
104 SIZE 3,41 ;
105 DEFAULT 1 ;
106 COLOR SCHEME 5, 6
107 @ 5,8 SAY "at step:"
108 @ 2,2 SAY "Select Factor:"
109 @ 8,24 GET m.action ;
110 PICTURE "g*HT \<Ok;\<Cancel" ;
111 SIZE 1,8,1 ;
112 DEFAULT 1 ;
113 VALID _qbqbnk6()
114
115 IF NOT EXISTS("bfact")
116 ACTIVATE WINDOW bfact
117 ENDIF
118
119 READ CYCLE MODAL
120
121 RELEASE WINDOW bfact
122 SELECT (m.curarea)
123
124 #REGION 0
125 IF m.talkstat = "ON"
126 SET TALK ON
127 ENDIF
128 IF m.compstat = "ON"
129 SET COMPATIBLE ON
130 ENDIF
131
132

```


BWFACT Cleanup Code

```

133 *
134 *
135 *
136 *
137 *
138 *
139 *
140 #REGION 1
141
142 RETURN m.text
143
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 *
152 *
153 *
154 *
155 *
156 *
157 *
158 FUNCTION _q8q0nqka6 && m.action VALID
159 #REGION 1
160 IF m.action = 1
161 m.text = array1[m.f1,1] + u + array2[m.f2,1]
162 ELSE
163 m.text = ""
164 ENDIF
165 *: EOF: BWFACT.ACT

```

_q8q0nqka6 m.action VALID

Function Origin:

From Screen: BWFACT, Record Number: 6
 Variable: m.action
 Called By: VALID Clause
 Object Type: Push Button
 Snippet Number: 1

08/24/93	DISPLAY.PRG	10:23:08
Author's Name		
Copyright (c) 1993 Company Name		
Address		
City,	Zip	
Description: This program was automatically generated by GENSCRN.		

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

PARAMETERS string

DISPLAY/MS-DOS Setup Code - SECTION 1

```

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

```

```

IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS
m.currarea = SELECT()

```

MS-DOS Window definitions

```

IF NOT WEXIST("w_hmresult") ;
OR UPPER(UTITLE("w_hmresult")) = "w_hmresult.pjx" ;
OR UPPER(UTITLE("w_hmresult")) = "w_hmresult.scx" ;
OR UPPER(UTITLE("w_hmresult")) = "w_hmresult.mnx" ;
OR UPPER(UTITLE("w_hmresult")) = "w_hmresult.prg" ;
OR UPPER(UTITLE("w_hmresult")) = "w_hmresult.frx" ;
OR UPPER(UTITLE("w_hmresult")) = "w_hmresult.qpr" ;
DEFINE WINDOW w_hmresult ;
FROM INT((SCOL()-20)/2), INT((SCOL()-77)/2) ;
TO INT((SCOL()-20)/2)+19, INT((SCOL()-77)/2)+76 ;
TITLE "Hazard Material Life Cycle Cost" ;
NOFLOAT ;
NOCLOSE ;
SHADOW ;
nominimize ;
DOUBLE ;
COLOR SCHEME 1
ENDIF

```

DISPLAY.ACT 12-1-93 11:32a

DISPLAY/MS-DOS Setup Code - SECTION 2

```

#REGION 1
m.action = 1
U = "u"
m.text = m.string

```

DISPLAY/MS-DOS Screen Layout

```

#REGION 1
IF WVISIBLE("w_hmresult")
ACTIVATE WINDOW w_hmresult SAME
ELSE
ACTIVATE WINDOW w_hmresult NOSHOW
ENDIF
@ 1,2 EDIT m.text ;
SIZE 14,72,0 ;
DEFAULT " " ;
SCROLL ;
NOMODIFY

```

```

@ 16,31 GET m.action ;
PICTURE "PRINT \<Print;\<Quit" ;
SIZE 1,7,3 ;
DEFAULT 1 ;
VALID _qindowp2j()

```

```

IF NOT WVISIBLE("w_hmresult")
ACTIVATE WINDOW w_hmresult
ENDIF

```

```

READ CYCLE MODAL
RELEASE WINDOW w_hmresult
SELECT (m.currarea)

```

```

#REGION 0
IF m.talkstat = "ON"
SET TALK ON
ENDIF
IF m.compstat = "ON"
SET COMPATIBLE ON
ENDIF

```

_qindowp2j m.action VALID
Function Origin:
From Platform: MS-DOS
From Screen: DISPLAY, Record Number: 3

Page 1 of 2

```

133 *
134 *
135 *
136 *
137 *
138 *
139 FUNCTION_qinOm9e2j  && m.action VALID
140 #REGION 1
141 IF m.action = 1
142 DO w_print.spr WITH m.text
143 ENDIF
144 *: EOF: DISPLAY.ACT

```

Variable: m.action
 Called By: VALID Clause
 Snippet Number: 1

```

1  * *****
2  *
3  * Procedure file: C:\HAZMAT\GHW\WORK\DP.PRG
4  *
5  * System: Hazardous Material Life-Cycle Cost Model
6  * Author: Naval Health Research Center
7  * Copyright (c) 1993, Naval Health Research Center
8  * Last modified: 09/10/93      8:12
9  *
10 * Set by: HMTAB.SPR
11 * : HMSC.PRG
12 * : HMSCS.PRG
13 * : SUBCOMPUT
14 * : SETUPBOOT()
15 * : GETSTR()
16 * : Q7000E18()
17 * : FILEFIND()
18 * : Q800INGJH()
19 * : Q800INGTV()
20 * : Q1POVASYA()
21 * : Q1POVAT9X()
22 * : Q1POVATLC()
23 *
24 *
25 * Documented 12/01/93 at 11:32 FoxDoc version 2.10f
26 * *****
27 *
28 * Date: 08/04/92 10:01:15
29 * Program Name: Question.prg
30 * Author's Name: Hoa Le Ly
31 *
32 * Copyright (c) 1992 Company Name: NHRC
33 * Department: Code 22
34 * San Diego, CA 92138 - 5122
35 * Description: This program emulate $PIECE of MUMPS function. Which
36 * => return the
37 * the portion of string which is bounded by the characters in deli
38 * miter. If both
39 * expr and expr2 are present, the value returned includes all char
40 * => actors from
41 * the expr1-1th occurrence of delimiter, up to but not including
42 * the expr2th
43 * occurrence of delimiter. If expr2 is not present. it is assumed
44 * => to have the same
45 * value as expr. If expr1 is not present. Then its value is assu
46 * me to be 1
47 * SYNTAX: DP(string, delimiter[,expr[,expr2]])
48 * PARAMETER: string: Character expression which character extract fro
49 * m
50 * delimiter: Character to delimiter
51 * expr: start of number occurrence of delimiter
52 * expr2: number of occurrence delimiter
53 * eg: string = "last, first age date of birth"
54 * ?dp(string,"-",1) ==> last, first
55 * ?dp(string,"-",1,2) ==> last, first age
56 * *****
57 *
58 * PARAMETER ms,mm,mp,mp2
59 * PRIVATE ALL
60 * DO CASE
61 * -CASE PARAMETER() < 2
62 * -RETURN ""
63 * -CASE PARAMETER() = 2

```

```

56 mp = 1
57 mp2 = -1
58 -CASE PARAMETER() = 3
59 -IF TYPE("mp") != ""
60 -RETURN ""
61 -ELSE
62 -IF mp < 1
63 -RETURN ""
64 -ENDIF
65 -ENDIF
66 mp2 = -1
67 -CASE PARAMETER() = 4
68 -IF TYPE("mp") != ""
69 -RETURN ""
70 -ENDIF
71 -IF TYPE("mp2") = ""
72 mp2 = 1 IF mp >= mp2, -1, mp2)
73 -ELSE
74 mp2 = -1
75 -ENDIF
76 -ENDCASE
77 -IF TYPE("ms") != "C"
78 ms = STR(ms)
79 -ENDIF
80 moccurs = OCCURS(mm,ms)
81 -IF moccurs = 0
82 => mp=1/(ms'(mm))
83 mstr = 1 IF mp = 1, ms, ""
84 -RETURN mstr
85 -ENDIF
86 mbegin = 1 IF mp = 1, 1, (AT(mm,ms,(mp-1)+1))
87 -IF mp2 = -1
88 mrend = AT(mm,ms,mp) - 1
89 -IF mrend < 0
90 -IF mbegin > 1
91 mrend = LEN(ms)
92 -ELSE
93 -RETURN ""
94 -ENDIF
95 -ENDIF
96 mrend = AT(mm,ms,mp2) - 1
97 mrend = 1 IF mrend < 0, LEN(ms), mrend)
98 -ENDIF
99 mstr = SUBSTR(ms, mbegin, (mrend - mbegin + 1))
100 RETURN mstr
101 * : EOF: DP.ACT

```

```

1  *****
2  *
3  * Procedure file: C:\HAZMAT\GHW\WORK\ERRMSG.PRG
4  *
5  *
6  * System: Hazardous Material Life-Cycle Cost Model
7  * Author: Naval Health Research Center
8  * Copyright (c) 1993, Naval Health Research Center
9  * Last modified: 09/10/93      8:12
10
11  Set by: HMTAB.SPR
12  : BACKUP.PRG
13  : YESNO.PRG
14  : HMSTEP.PRG
15  : HMGETPMD.PRG
16  : QKFOVR304()
17  : QKFOVR304()
18  : QKFOVR304()
19  : QKFOVR304()
20  : QKFOVR304()
21  : QKFOVR304()
22  : QKFOVR304()
23  : QKFOVR304()
24  : QKFOVR304()
25  : QKFOVR304()
26  : QKFOVR304()
27  : QKFOVR304()
28  : QKFOVR304()
29  : QKFOVR304()
30  : QKFOVR304()
31  : QKFOVR304()
32  : QKFOVR304()
33  : QKFOVR304()
34  : QKFOVR304()
35  : QKFOVR304()
36  : QKFOVR304()
37  : QKFOVR304()
38  : QKFOVR304()
39  : QKFOVR304()
40  : QKFOVR304()
41  : QKFOVR304()
42  : QKFOVR304()
43  : QKFOVR304()
44  : QKFOVR304()
45  : QKFOVR304()
46  : QKFOVR304()
47  : QKFOVR304()
48  : QKFOVR304()
49  : QKFOVR304()
50  : QKFOVR304()
51  : QKFOVR304()
52  : QKFOVR304()
53  : QKFOVR304()
54  : QKFOVR304()
55  : QKFOVR304()
56  : QKFOVR304()
57  : QKFOVR304()
58  : QKFOVR304()
59  : QKFOVR304()
60  : QKFOVR304()
61  : QKFOVR304()
62  : QKFOVR304()
63  : QKFOVR304()
64  : QKFOVR304()

```

```

65  *****
66  *
67  * IF TYPE("m.TimeLimit")="M"
68  * m.timeLimit=11F(TYPE("m.TimeLimit")="C",INT(VAL(m.timeLimit)),0)
69  *
70  *
71  * m.errmsg=11F(TYPE("m.errmsg")="C","m.errmsg")
72  *
73  * IF PARAMETERS()=2 AND EMPTY(m.errmsg) AND m.timeLimit>0
74  * RETURN
75  *
76  *
77  * * Get the length of the message to size window
78  * m.len=LEN(m.errmsg)
79  *
80  * * Set minimum length for the Press any key
81  * IF m.timeLimit = 0
82  * m.len=11F(m.len<28,28,m.len)
83  *
84  *
85  * IF m.len > 70
86  * m.errmsg=SUBSTR(m.errmsg,1,70)
87  *
88  *
89  * * Find beginning/ending of the window
90  * m.begin=40-(INT(m.len/2)+1)
91  * m.end=40+(INT(m.len/2)+1)
92  *
93  * * Remember the current window status
94  * m.oldwindow=11F(WOUPUT()=errmsg,"",WOUPUT())
95  *
96  *
97  * IF NOT WEXIST("ErrMsg")
98  * DEFINE WINDOW errmsg ;
99  * FROM 1,m.begin ;
100 * TO 4,m.end ;
101 * SHADOW ;
102 * DOUBLE ;
103 * COLOR SCHEME 5
104 *
105 *
106 * ACTIVATE WINDOW errmsg
107 *
108 * * Print the message centered in the window
109 *
110 * CLEAR
111 * @ 0, ( WCOL() - LEN( m.errmsg ) )/2 SAY m.errmsg
112 *
113 * IF m.timeLimit = 0
114 * m.pressedkey = "Press any key to continue"
115 * @ 1, ( WCOL() - LEN( m.pressedkey ) )/2 SAY m.pressedkey
116 *
117 *
118 * * Wait for the number of seconds in m.Timeout
119 * * A value of 0 will wait forever
120 * WAIT "" TIMEOUT m.timeLimit
121 *
122 * * Close Window
123 * RELEASE WINDOW errmsg
124 *
125 * * If there was no output window originally
126 * IF EMPTY( m.oldwindow )
127 * * Send future output back to the screen
128 * ACTIVATE SCREEN
129 *
130 * ELSE

```

```
131
132 * Return output to the original window
133 ACTIVATE WINDOW ( m.oldWindow )
134 ENDIF
135
136 SET TALK &saveTalk  ## Restore original TALK setting
137 SET COLOR SET TO ## RESTORE OLD COLOR SET
138 RETURN
139 *: EOF: ERRMSG.ACT
```

```

1 *****
2 => *****
3 *: Procedure file: C:\HAZMAT\GMM\WORK\HNGETPWD.PRG
4 *:
5 *: System: Hazardous Material Life-Cycle Cost Model
6 *: Author: Naval Health Research Center
7 *: Copyright (c) 1993, Naval Health Research Center
8 *: Last modified: 09/10/93 8:12
9 *:
10 *: Set by: HMINIT.PRG
11 *: : HMINIT.PRG
12 *:
13 *: Calls: ERRMSG.PRG
14 *:
15 *: Documented 11/24/93 at 07:55 FoxDoc version 2.10f
16 *: *****
17 => *****
18 *:
19 *: * Validation the password. Is boolean returnable
20 *:
21 *****
22 *****
23 *****
24 *****
25 *****
26 *****
27 *****
28 *****
29 *****
30 *****
31 *****
32 *****
33 *****
34 *****
35 *****
36 *****
37 *****
38 *****
39 *****
40 *****
41 *****
42 *****
43 *****
44 *****
45 *****
46 *****
47 *****
48 *****
49 *****
50 *****
51 *****
52 *****
53 *****
54 *****
55 *****
56 *****
57 *****
58 *****
59 *****
60 *****
61 *****

```

```

62 *****
63 *****
64 *****
65 *****
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****
89 *****
90 *****
91 *****
92 *****
93 *****
94 *****
95 *****
96 *****
97 *****
98 *****
99 *****
100 *****
101 *****
102 *****
103 *****
104 *****
105 *****
106 *****
107 *****
108 *****
109 *****
110 *****
111 *****
112 *****
113 *****
114 *****
115 *****
116 *****
117 *****
118 *****
119 *****
120 *****
121 *****
122 *****
123 *****

```

```

1  *:*:*****
2  *:*:*****
3  *:*: Procedure file: C:\HAZMAT\GHW\WORK\HMINIT.PRG
4  *:*:
5  *:*: System: Hazardous Material Life-Cycle Cost Model
6  *:*: Author: Naval Health Research Center
7  *:*: Copyright (c) 1993, Naval Health Research Center
8  *:*: Last modified: 09/10/93 8:11
9  *:*:
10 *:*: Procs & Frcts: MYHANDLER()
11 *:*: : _QUIT()
12 *:*:
13 *:*: Calls: HMGTPWD.PRG
14 *:*: : BIGCHARS.PRG
15 *:*: : HMENU.MPR
16 *:*: : MYHANDLER()
17 *:*: (function in HMINIT.PRG)
18 *:*:
19 *:*: Documented 12/01/93 at 11:31 FoxDoc version 2.10f
20 *:*:*****
21 *:*:*****
22 *:*: FUNCTION:HMINIT.PRG
23 *:*: PURPOSE: DRIVER FOR HMENU
24 *:*: NOTE: 3 CHANCES FOR ACCESSING ALLOWED.
25 *:*: PROCEDURES CALLED:
26 *:*: SIDE EFFECTS: NONE KNOWN.
27 *:*: REFERENCE: C:\EXAMPLE\PRGS\EX2.PRG
28 *:*: CREATED: 06/12/92 AL
29 *:*: MODIFIED: 06/14/92 AL
30 *:*:
31 *:*:*****
32 *:*:*****
33 *:*:*****
34 *:*:*****
35 *:*:*****
36 *:*:*****
37 *:*:*****
38 *:*:*****
39 *:*:*****
40 *:*:*****
41 *:*:*****
42 *:*:*****
43 *:*:*****
44 *:*:*****
45 *:*:*****
46 *:*:*****
47 *:*:*****
48 *:*:*****
49 *:*:*****
50 *:*:*****
51 *:*:*****
52 *:*:*****
53 *:*:*****
54 *:*:*****
55 *:*:*****
56 *:*:*****
57 *:*:*****
58 *:*:*****
59 *:*:*****
60 *:*:*****

```

```

61 *SET HELP TO HHHELP.DBF && Change help database
62 *SET HELP TO && TO DEFAULT HELP
63 SET SYSMENU TO DEFAULT && DEFAULT SYSTEM MENU
64
65 && End of init_app()
66 *****
67
68 *
69 * Get password and print to screen*
70 *
71 *****
72
73 m.pass=UPPER(hmgtpwd("")) && Get password
74
75 IF m.pass != "SYSHAZ"
76 m.pass="HAZMAT"
77 ENDIF
78
79 m.appbr=IF(m.pass="SYSHAZ","HMLCCN","HMLCCN") && Set title of applic
=> action
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124

```



```
125 *| Function: _QUIT()
126 *|
127 *| Called by: HMENU.MPR
128 *|
129 *|*****
=> *****
123 FUNCTION _quit
124   dropped = 1;
125   CLEAR READ ALL
126   RETURN
127   *: EOF: HMJINIT.ACT
```

```

67 #REGION 1
68 *#SECTION 1
69 *SET UP PARAMETERS
70 m.hname=" "
71 m.value = " "
72 *
73 *
74 *
75 *
76 *
77 *
78
79 #REGION 1
80 IF WVISBLE("hmlu")
81     ACTIVATE WINDOW hmlu SAME
82 ELSE
83     ACTIVATE WINDOW hmlu NOSHOW
84 ENDIF
85 @ 2,4 SAY "MATERIAL:"
86 @ 2,16 GET m.hname ;
87 SIZE 1,43 ;
88 DEFAULT " " ;
89 PICTURE "g1" ;
90 VALID _q7d0o1e18(
91 @ 3,17 SAY "(Enter M1IN, Brand Name or '?' for Help)"
92 @ 6,25 GET m.okcan ;
93 PICTURE "g2HT \<Ok;\<Cancel" ;
94 SIZE 1,81 ;
95 DEFAULT 2 ;
96 VALID _q7d0o1eub(
97
98 IF NOT WVISBLE("hmlu")
99     ACTIVATE WINDOW hmlu
100 ENDIF
101
102 READ CYCLE MODAL
103
104 RELEASE WINDOW hmlu
105 SELECT (m.curarea)
106
107 #REGION 0
108 IF m.talkstat = "ON"
109     SET TALK ON
110 ENDIF
111 IF m.compstat = "ON"
112     SET COMPATIBLE ON
113 ENDIF
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
```

```

1  *
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 #REGION 0
19 REGIONAL m.curraea, m.talkstat, m.compstat
20
21 IF SET("TALK") = "ON"
22   SET TALK OFF
23   m.talkstat = "ON"
24 ELSE
25   m.talkstat = "OFF"
26 ENDIF
27 m.compstat = SET("COMPATIBLE")
28 SET COMPATIBLE FOXPLUS
29
30 m.curraea = SELECT()
31 IF USED("hmat")
32   SELECT hmat
33   SET ORDER TO 0
34 ELSE
35   SELECT 0
36   USE (LOGFILE("hmat.dbf","DBF","Where is hmat?"));
37   AGAIN ALIAS hmat ;
38   ORDER 0
39 ENDIF
40 U = " "
41
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *

```

Window definitions

```

66
67 IF NOT WEXIST("hmlu")
68   DEFINE WINDOW hmlu ;
69   FROM INT((SCROW()-9)/2),INT((SCOL()-66)/2) ;
70   TO INT((SCROW()-9)/2)+8,INT((SCOL()-66)/2)+65 ;
71   FLOAT ;
72   NOCLOSE ;
73   SHADOW ;
74   DOUBLE ;
75   COLOR SCHEME 1
76 ENDIF
77
78 *
79 *
80 *
81 *
82 *
83 *
84 *
85 *
86 *
87 *
88 *
89 *
90 *
91 *
92 *
93 *
94 *
95 *
96 *
97 *
98 *
99 *
100 *

```

HMLU Setup Code - SECTION 2

```

133 DIMENSION hnm(1)
134
135 IF LEN(m.name) = 9 AND TYPE(m.name) = "M"
136 SELECT DISTINCT ALLTRIM(hmat.hmatname) + IIF(EMPTY(hmat.mfg), " [
=> " + ALLTRIM(hmat.mfg) + "] ", "")
137 hmat.hmatname, hmat.hmatid;
138 FROM hmat;
139 WHERE UPPER(hmat.niin) = UPPER(m.name);
140 INTO ARRAY hnm
141
142 ELSE
143 SELECT DISTINCT ALLTRIM(hmat.hmatname) + IIF(EMPTY(hmat.mfg), " [
=> " + ALLTRIM(hmat.mfg) + "] ", "")
144 hmat.hmatname, hmat.hmatid;
145 FROM hmat;
146 WHERE UPPER(hmat.hmatname) = UPPER(m.name);
147 INTO ARRAY hnm
148 ENDIF
149
150 m.result=""
151 IF NOT EMPTY(hnm[1])
152 m.ans=chooser(ahm,"Select a Material")
153 m.temp=ASUSCRIPT(hnm,ASCAN(hnm,m.ans),1)
154 m.hmid = hnm[m.temp,3]
155 m.result = ALLTRIM(hnm[m.temp,2]) + U + ALLTRIM(STR(m.hmid))
156 ELSE
157 zerrmsg(m.name + " was not found",1)
158 ENDIF
159
160 *m.hname=m.ans
161 RETURN m.result
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196

```

_q70001E18 m.hname VALID

Function Origin:

From Screen: hmlu, Record Number: 2
 Variable: m.hname
 Called By: VALID Clause
 Object Type: Field
 Snippet Number: 1

```

176 FUNCTION _q7d001e18 @@ m.hname VALID
177 m.value = ""
178 IF NOT EMPTY(m.hname)
179 m.hname=ALLTRIM(m.hname)
180 m.hname=IIF(m.hname="?", "", m.hname)
181 m.value=get_hname(m.hname)
182 m.hname = cp(m.value,u,1)
183
184
185
186
187
188
189
190
191
192
193
194
195
196

```

_q70001E18 m.okcan VALID

Function Origin:

From Screen: hmlu, Record Number: 4
 Variable: m.okcan
 Called By: VALID Clause
 Object Type: Push Button
 Snippet Number: 2

```

197
198
199
200 FUNCTION _q7d001e1b @@ m.okcan VALID
201 #REGION 1
202 DO CASE
203 CASE m.okcan=1
204 RETURN m.value
205
206 CASE m.okcan=2
207 m.value=""
208 ENDCASE
209 *: EOF: HMLU.ACT

```

```

1  * *****
2  * Procedure file: C:\HAZMAT\GHH\WORK\HMMSDS.PRG
3  *
4  *
5  * System: Hazardous Material Life-Cycle Cost Model
6  * Author: Naval Health Research Center
7  * Copyright (c) 1993, Naval Health Research Center
8  * Last modified: 09/10/93 8:12
9  *
10 * Set by: HMENU.MPR
11 *
12 * Calls: OPENFILE() (function in HMSC.PRG)
13 * : DP.PRG
14 * : HMLU.PRG
15 * : HMVIEW() (function in HMREF.PRG)
16 *
17 * Documented 12/01/93 at 11:32 FoxDoc version 2.10f
18 * *****
19 * hmsds.prg gp 8/9/93
20 * show mds for selected product
21 *
22 IF SET("TALK") = "ON"
23 SET TALK OFF
24 m.talkstat = "ON"
25 ELSE
26 m.talkstat = "OFF"
27 ENDIF
28 m.oldproc = SET("PROC")
29 m.oldscape = SET("ESCAPE")
30 SET ESCAPE OFF
31 SET PROC TO hmsc.prg
32 mfind = openfile("hmat")
33 IF mfind
34 RETURN
35 ENDIF
36
37 m.quit = .F.
38
39 *****
40 * SELECT HAZARDOUS MATERIAL
41 *****
42 DO WHILE m.quit
43 * now return hname - hmid 9/7/93 gp
44 m.hname = op(hmat, " ", 1)
45 IF !EMPTY(m.hname)
46 = hmview(m.hname)
47 m.quit = .F.
48 ELSE
49 m.quit = .T.
50 ENDIF
51 ENDDO
52
53 SET ESCAPE &oldscape
54 SET TALK &talkstat
55 IF !EMPTY(m.oldproc)
56 SET PROC TO &oldproc
57 ELSE
58 SET PROC TO
59 ENDIF
60
61 RETURN
62
63 *
64 *

```

```

65 * Display the text to screen
66 *
67 * *****
68 *
69 *
70 *
71 * Function: HMVIEW()
72 * Called by: HMREF.PRG
73 * : HMMSDS.PRG
74 *
75 * Calls: NEWLINE() (function in HMAT.SPR)
76 * : MEMOWIN.PRG
77 *
78 * *****
79 *
80 FUNCTION hmview
81 PARAMETER name
82
83 IF PARAMETER() = 0
84 RETURN
85 ENDIF
86
87 SELECT hmat
88 SET ORDER TO TAG hmatname
89 GO TOP
90 SEEK name
91 IF FOUND()
92 SCATTER MEMVAR MEMO
93 m.alltext = m.mds
94 m.title = (m.nlin) + " " + (UPPER(ALLTRIM(m.hmatname))) + " ";
95 + (UPPER(ALLTRIM(m.mfg)))
96 * m.title = UPPER(alltrim(name))
97 = memowin(m.alltext, m.title)
98 ENDIF
99 RETURN
100 *: EOF: HMMSDS.ACT

```

```

1  * *****
2  * Procedure file: C:\HAZMAT\GHM\WORK\HREF.PRG
3  *
4  * System: Hazardous Material Life-Cycle Cost Model
5  * Author: Naval Health Research Center
6  * Copyright (c) 1993, Naval Health Research Center
7  * Last modified: 09/10/93 8:12
8  *
9  * Procs & Fncts: HNVIEW()
10 *
11 *
12 * Calls: OPENFILE() (function in HMSC.PRG)
13 * : HMLU.PRG
14 * : HNVIEW() (function in HREF.PRG)
15 *
16 * Documented 12/01/93 at 11:32 FoxDoc version 2.10f
17 * *****
18 *
19 IF SET("TALK") = "ON"
20 SET TALK OFF
21 m.talkstat = "ON"
22 ELSE
23 m.talkstat = "OFF"
24 ENDIF
25 m.oldproc = SET("PROC")
26 m.oldscape = SET("ESCAPE")
27 SET ESCAPE OFF
28 SET PROC TO hmsc.prg
29 mfind = openfile("hmat")
30 IF mfind
31 RETURN
32 ENDF
33
34 m.quit = .F.
35
36 *****
37 * SELECT HAZARDOUS MATERIAL
38 *****
39 DO WHILE m.quit
40 m.hname = hmlu()
41 IF !EMPTY(m.hname)
42 = hmvview(m.hname)
43 m.quit = .F.
44 ELSE
45 m.quit = .T.
46 ENDF
47 ENDDO
48
49 SET ESCAPE &oldscape
50 SET TALK &talkstat
51 IF !EMPTY(m.oldproc)
52 SET PROC TO &oldproc
53 ELSE
54 SET PROC TO
55 ENDF
56
57 RETURN
58
59 *
60 * Display the text to screen
61 *
62 *
63 * *****

```

```

76 *
77 * Function: HNVIEW()
78 * Called by: HREF.PRG
79 * : HMSCDS.PRG
80 *
81 * Calls: NEWLINE() (function in HMAT.SPR)
82 * : MEMOWIN.PRG
83 *
84 * *****
85 *
86 FUNCTION hmvview
87 PARAMETER name
88
89 IF PARAMETER() = 0
90 RETURN
91 ENDF
92
93 SELECT hmat
94 SET ORDER TO TAG hmatname
95 GO TOP
96 SEEK name
97 IF FOUND()
98 SCATTER MEMVAR MEMO
99 nl = CHR(10) + CHR(13)
100 m.alltext = newline(m.alltext);
101 + nl + m.syntrade;
102 m.alltext = newline(m.alltext);
103 + IIF(!EMPTY(m.com uses), "COMMONS USES" + nl + m.com_uses, "")
104 m.alltext = newline(m.alltext);
105 + IIF(!EMPTY(m.hmdescript), "DESCRIPTIONS" + nl + m.hmdescript, "")
106 m.alltext = newline(m.alltext);
107 + IIF(!EMPTY(m.chem.phy), "CHEMICAL/PHYSICAL PROPERTIES";
108 + nl + m.chem.phy, "");
109 m.alltext = newline(m.alltext);
110 + IIF(!EMPTY(m.occ exp), "OCCUPATIONAL EXPOSURES";
111 + nl + m.occ exp, "");
112 m.alltext = newline(m.alltext);
113 + IIF(!EMPTY(m.exp limits), "EXPOSURE LIMITS";
114 + nl + m.exp limits, "");
115 m.alltext = newline(m.alltext);
116 + IIF(!EMPTY(m.health_haz), "HEALTH HAZARDS" + nl + m.health_haz, "");
117 m.alltext = newline(m.alltext);
118 + IIF(!EMPTY(m.med surv), "MEDICAL SURVEILLANCE";
119 + nl + m.med surv, "");
120 m.alltext = newline(m.alltext);
121 + IIF(!EMPTY(m.spec tests), "SPECIAL TESTS" + nl + m.spec_tests, "");
122 m.alltext = newline(m.alltext);
123 + IIF(!EMPTY(m.ppe treat), "PERSONAL PROTECTIVE EQUIPMENT";
124 + nl + m.ppe treat, "");
125 m.alltext = newline(m.alltext);
126 + IIF(!EMPTY(m.treatment), "TREATMENTS" + nl + m.treatment, "");
127 m.title = UPPER(ALLTRIM(m.hmatname))
128 = memowin(m.alltext, m.title)
129 ENDF
130 RETURN
131
132 *****
133 *
134 * *****

```

```

134 *!      Function: NEMLINE()
135 *!
136 *!      Called by: HMVIEW()      (function in HMREF.PRG)
137 *!
138 *!*****
=> *****
132      FUNCTION newline
133      *****
134      PARAMETER TEXT
135      nl = CHR(10) + CHR(13)
136      -IF EMPTY(m.text)
137      {
138      -ELSE
139      { m.text = m.text + nl + nl
140      -ENDIF
141      RETURN m.text
142      *: EOF: HMREF.ACT

```

```

1  * *****
2  * Procedure file: C:\HAZMAT\GHW\WORK\HMSC.PRG
3  *
4  *
5  * System: Hazardous Material Life-Cycle Cost Model
6  * Author: Naval Health Research Center
7  * Copyright (c) 1993, Naval Health Research Center
8  * Last modified: 09/10/93 8:12
9  *
10 * Procs & Fncts: OPENFILE()
11 * : POPUPSHOW()
12 * : POPUPHIDE()
13 * : DEL_HMSC()
14 * : CLOSEFILE
15 * : FILEFIND()
16 *
17 * Set by: HMENU.MPR
18 *
19 * Calls: HMSCEN.PRG
20 * : DP.PRG
21 * : HNSTEP.PRG
22 * : DEL_HMSC()
23 * : YESNO.PRG
24 * : HMCMP.SPR
25 * : CLOSEFILE
26 *
27 *
28 * Documented 12/01/93 at 11:32 FoxDoc version 2.10f
29 * *****
30 * PUBLIC mtalk, mesc, u
31 * IF SET("TALK") = "ON"
32 *   SET TALK OFF
33 *   mtalk = "ON"
34 *   ENDIF
35 *   mesc = SET("ESCAPE")
36 *   SET ESCAPE OFF
37 *   u = ""
38 *   m.quitflag = .F.
39 *   mdata = hmscen()
40 *   mevent = dp(mdata,u,1)
41 *   m.hmscid = INIT(VAL(dp(mdata,u,2)))
42 *   m.hmscname = dp(mdata,u,3)
43 *
44 * DO CASE
45 * CASE mevent = "C" OR EMPTY(mevent)
46 * RETURN
47 * CASE mevent = "N"
48 * m.adding = .T.
49 * m.quitflag = hmsstep(m.adding, m.hmscid, m.hmscname)
50 * CASE mevent = "R"
51 * m.adding = .F.
52 * DO hmsstep WITH m.adding, m.hmscid, m.hmscname
53 * m.quitflag = .F.
54 * CASE mevent = "D"
55 * = del_hmsc(m.hmscid)
56 * m.quitflag = .T.
57 * OTHERWISE
58 * RETURN
59 * END CASE
60 * IF !m.quitflag
61 * m.compute = yesno("Do Computation Now?", "Yes", "No")
62 * IF m.compute
63 * DO hmscomp.spr WITH m.hmscid, m.hmscname
64 * ENDIF

```

```

65 DO closefile
66 RETURN
67
68 *****
69 * *****
70 * Procedure: CLOSEFILE
71 *
72 * Called by: HMSC.PRG
73 *
74 *****
75
76 PROCEDURE closefile
77 *****
78 CLOSE DATABASE
79 IF EMPTY(mesc)
80 SET ESCAPE &mesc
81 ENDIF
82 IF EMPTY(mtalk)
83 SET TALK &mtalk
84 ENDIF
85 RETURN
86
87 *****
88
89 *****
90 Function: DEL_HMSC()
91
92 Called by: HMSC.PRG
93
94 Calls: POPUPSHOW() (function in HMSC.PRG)
95 : OPENFILE() (function in HMSC.PRG)
96 : POPUPHIDE()
97
98 *****
99
100 FUNCTION del_hmsc
101 *****
102 PARAMETER id
103 wpop = popupshow("Deleting...")
104 filesuccess = openfile("hmsstep")
105 SET ORDER TO TAG hmscid
106 IF filesuccess
107 DELETE FOR hmscid = m.id
108 PACK
109 ENDIF
110 =popuphide(wpop)
111 RETURN
112 *
113 *
114 * Find the location of the file and file path will update into new pa
115 * if the file path has not in the foxpro path yet.
116 *
117 *
118 *****
119
120 *****
121
122 *****
123
124 *****
125
126 *****
127
128 *****
129
130 *****
131
132 *****
133
134 *****
135
136 *****
137
138 *****
139
140 *****
141
142 *****
143
144 *****
145
146 *****
147
148 *****
149
150 *****
151
152 *****
153
154 *****
155
156 *****
157
158 *****
159
160 *****
161
162 *****
163
164 *****
165
166 *****
167
168 *****
169
170 *****
171
172 *****
173
174 *****
175
176 *****
177
178 *****
179
180 *****
181
182 *****
183
184 *****
185
186 *****
187
188 *****
189
190 *****
191
192 *****
193
194 *****
195
196 *****
197
198 *****
199
200 *****
201
202 *****
203
204 *****
205
206 *****
207
208 *****
209
210 *****
211
212 *****
213
214 *****
215
216 *****
217
218 *****
219
220 *****
221
222 *****
223
224 *****
225
226 *****
227
228 *****
229
230 *****
231
232 *****
233
234 *****
235
236 *****
237
238 *****
239
240 *****
241
242 *****
243
244 *****
245
246 *****
247
248 *****
249
250 *****
251
252 *****
253
254 *****
255
256 *****
257
258 *****
259
260 *****
261
262 *****
263
264 *****
265
266 *****
267
268 *****
269
270 *****
271
272 *****
273
274 *****
275
276 *****
277
278 *****
279
280 *****
281
282 *****
283
284 *****
285
286 *****
287
288 *****
289
290 *****
291
292 *****
293
294 *****
295
296 *****
297
298 *****
299
300 *****
301
302 *****
303
304 *****
305
306 *****
307
308 *****
309
310 *****
311
312 *****
313
314 *****
315
316 *****
317
318 *****
319
320 *****
321
322 *****
323
324 *****
325
326 *****
327
328 *****
329
330 *****
331
332 *****
333
334 *****
335
336 *****
337
338 *****
339
340 *****
341
342 *****
343
344 *****
345
346 *****
347
348 *****
349
350 *****
351
352 *****
353
354 *****
355
356 *****
357
358 *****
359
360 *****
361
362 *****
363
364 *****
365
366 *****
367
368 *****
369
370 *****
371
372 *****
373
374 *****
375
376 *****
377
378 *****
379
380 *****
381
382 *****
383
384 *****
385
386 *****
387
388 *****
389
390 *****
391
392 *****
393
394 *****
395
396 *****
397
398 *****
399
400 *****
401
402 *****
403
404 *****
405
406 *****
407
408 *****
409
410 *****
411
412 *****
413
414 *****
415
416 *****
417
418 *****
419
420 *****
421
422 *****
423
424 *****
425
426 *****
427
428 *****
429
430 *****
431
432 *****
433
434 *****
435
436 *****
437
438 *****
439
440 *****
441
442 *****
443
444 *****
445
446 *****
447
448 *****
449
450 *****
451
452 *****
453
454 *****
455
456 *****
457
458 *****
459
460 *****
461
462 *****
463
464 *****
465
466 *****
467
468 *****
469
470 *****
471
472 *****
473
474 *****
475
476 *****
477
478 *****
479
480 *****
481
482 *****
483
484 *****
485
486 *****
487
488 *****
489
490 *****
491
492 *****
493
494 *****
495
496 *****
497
498 *****
499
500 *****
501
502 *****
503
504 *****
505
506 *****
507
508 *****
509
510 *****
511
512 *****
513
514 *****
515
516 *****
517
518 *****
519
520 *****
521
522 *****
523
524 *****
525
526 *****
527
528 *****
529
530 *****
531
532 *****
533
534 *****
535
536 *****
537
538 *****
539
540 *****
541
542 *****
543
544 *****
545
546 *****
547
548 *****
549
550 *****
551
552 *****
553
554 *****
555
556 *****
557
558 *****
559
560 *****
561
562 *****
563
564 *****
565
566 *****
567
568 *****
569
570 *****
571
572 *****
573
574 *****
575
576 *****
577
578 *****
579
580 *****
581
582 *****
583
584 *****
585
586 *****
587
588 *****
589
590 *****
591
592 *****
593
594 *****
595
596 *****
597
598 *****
599
600 *****
601
602 *****
603
604 *****
605
606 *****
607
608 *****
609
610 *****
611
612 *****
613
614 *****
615
616 *****
617
618 *****
619
620 *****
621
622 *****
623
624 *****
625
626 *****
627
628 *****
629
630 *****
631
632 *****
633
634 *****
635
636 *****
637
638 *****
639
640 *****
641
642 *****
643
644 *****
645
646 *****
647
648 *****
649
650 *****
651
652 *****
653
654 *****
655
656 *****
657
658 *****
659
660 *****
661
662 *****
663
664 *****
665
666 *****
667
668 *****
669
670 *****
671
672 *****
673
674 *****
675
676 *****
677
678 *****
679
680 *****
681
682 *****
683
684 *****
685
686 *****
687
688 *****
689
690 *****
691
692 *****
693
694 *****
695
696 *****
697
698 *****
699
700 *****
701
702 *****
703
704 *****
705
706 *****
707
708 *****
709
710 *****
711
712 *****
713
714 *****
715
716 *****
717
718 *****
719
720 *****
721
722 *****
723
724 *****
725
726 *****
727
728 *****
729
730 *****
731
732 *****
733
734 *****
735
736 *****
737
738 *****
739
740 *****
741
742 *****
743
744 *****
745
746 *****
747
748 *****
749
750 *****
751
752 *****
753
754 *****
755
756 *****
757
758 *****
759
760 *****
761
762 *****
763
764 *****
765
766 *****
767
768 *****
769
770 *****
771
772 *****
773
774 *****
775
776 *****
777
778 *****
779
780 *****
781
782 *****
783
784 *****
785
786 *****
787
788 *****
789
790 *****
791
792 *****
793
794 *****
795
796 *****
797
798 *****
799
800 *****
801
802 *****
803
804 *****
805
806 *****
807
808 *****
809
810 *****
811
812 *****
813
814 *****
815
816 *****
817
818 *****
819
820 *****
821
822 *****
823
824 *****
825
826 *****
827
828 *****
829
830 *****
831
832 *****
833
834 *****
835
836 *****
837
838 *****
839
840 *****
841
842 *****
843
844 *****
845
846 *****
847
848 *****
849
850 *****
851
852 *****
853
854 *****
855
856 *****
857
858 *****
859
860 *****
861
862 *****
863
864 *****
865
866 *****
867
868 *****
869
870 *****
871
872 *****
873
874 *****
875
876 *****
877
878 *****
879
880 *****
881
882 *****
883
884 *****
885
886 *****
887
888 *****
889
890 *****
891
892 *****
893
894 *****
895
896 *****
897
898 *****
899
900 *****
901
902 *****
903
904 *****
905
906 *****
907
908 *****
909
910 *****
911
912 *****
913
914 *****
915
916 *****
917
918 *****
919
920 *****
921
922 *****
923
924 *****
925
926 *****
927
928 *****
929
930 *****
931
932 *****
933
934 *****
935
936 *****
937
938 *****
939
940 *****
941
942 *****
943
944 *****
945
946 *****
947
948 *****
949
950 *****
951
952 *****
953
954 *****
955
956 *****
957
958 *****
959
960 *****
961
962 *****
963
964 *****
965
966 *****
967
968 *****
969
970 *****
971
972 *****
973
974 *****
975
976 *****
977
978 *****
979
980 *****
981
982 *****
983
984 *****
985
986 *****
987
988 *****
989
990 *****
991
992 *****
993
994 *****
995
996 *****
997
998 *****
999
1000 *****

```

```

134 *!      Calls: DP.PRG
135 *!
136 *!*****
=> *****
128 FUNCTION filefind
129 *****
130 PARAMETER mfilename
131 PRIVATE mflag
132 mflag = .T.
133 IF PARAMETER() = 0
134 RETURN .F.
135 ENDIF
136 ON ERROR RETURN .F.
137 IF IFILE(mfilename)
138 mfile = dp(mfilename,"",1)
139 next = dp(mfilename,"",2)
140 =LOCFILE(mfile,next,"where is the " + mfilename + "?")
141 ENDIF
142 ON ERROR
143 RETURN mflag
144
145 *****
=> *****
160 *!      Function: OPENFILE()
161 *!
162 *!      Called by: HMSTEP.PRG
163 *!                  : HMREF.PRG
164 *!                  : HMMSDS.PRG
165 *!                  : OPEN
166 *!                  : DEL_HMSC()
167 *!
168 *!      Calls: FILEFIND()
169 *!
170 *!*****
=> *****
159 FUNCTION openfile
160 *****
161 PARAMETER FILE
162 IF filefind(FILE + ".dbf")
163 VALID = .T.
164 IF IUSED(FILE)
165 SELECT 0
166 USE (FILE)
167 ELSE
168 SELECT (FILE)
169 ENDIF
170 ELSE
171 VALID = .F.
172 ENDIF
173 RETURN VALID
174
175 *****
176 * display popup notice
177 *****
178
179 *****
=> *****
188 *!      Function: POPUPSHOW()
189 *!
190 *!      Called by: QLEOL6CBO()
191 *!                  : DEL_HMSC()
192 *!
193 *!*****

```

```

=> *****
187 FUNCTION popupshow
188 PARAMETERS errstr
189
190 IF NOT EXIST("w_papnote")
191 DEFINE WINDOW w_papnote
192 FROM INT((SROW()-8)/2), INT((SCOL()-36)/2)
193 TO INT((SROW()-8)/2)+7, INT((SCOL()-36)/2)+35
194 TITLE "One moment"
195 FLOAT
196 CLOSE
197 SHADOW
198 DOUBLE
199 COLOR SCHEME 5
200 ENDIF
201 IF WVISIBLE("w_papnote")
202 ACTIVATE WINDOW w_papnote SAME
203 ELSE
204 ACTIVATE WINDOW w_papnote NOSHOW
205 ENDIF
206 @ 1,1 SAY errstr SIZE 3,31
207
208 IF NOT WISIBLE("w_papnote")
209 ACTIVATE WINDOW w_papnote
210 ENDIF
211 RETURN ""
212
213 *****
=> *****
223 *!      Function: POPUPHIDE()
224 *!
225 *!      Called by: QLEOL6CBO()
226 *!                  : DEL_HMSC()
227 *!
228 *!      (function in HMCOMP.SPR)
229 *!      (function in HMISC.PRG)
230
231 *****
=> *****
222 FUNCTION popuphide
223 *****
224 PARAMETERS w
225 RELEASE WINDOW w_papnote
226 RETURN w
227 * EOF: HMISC.ACT

```


08/27/92	HMSCEN.PRG	08:42:06
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCRN.		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *

```

```

18 #REGION 0
19 REGIONAL m.curarea, m.talkstat, m.compstat
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

```

```

42 IF SET("TALK") = "ON"
43 SET TALK OFF
44 m.talkstat = "ON"
45 ELSE
46 m.talkstat = "OFF"
47 ENDIF
48 m.compstat = SET("COMPATIBLE")
49 SET COMPATIBLE FOXPLUS
50 m.curarea = SELECT()
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

Window definitions

```

67 IF NOT EXIST("w_hmscen")
68 DEFINE WINDOW w_hmscen ;
69 FROM INT((SCOL()-7)/2), INT((SCOL()-66)/2) ;
70 TO INT((SCOL()-7)/2)+6, INT((SCOL()-66)/2)+65 ;
71 TITLE "Cost Analysis" ;
72 NOFLOAT ;
73 NOCLOSE ;
74 SHADOW ;
75 DOUBLE ;
76 COLOR SCHEME 1
77 ENDIF
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132

```

HMSCEN Setup Code - SECTION 2

```

133 #REGION 1
134 m.hmscen = ""
135 m.hmscenid = 0
136 m.driven = ""
137 mevent = 1
138 mstatus = 1

```

```

67 u = ""
68 msel = SELECT()
69
70 IF IUSED("hmscen")
71 SELECT 0
72 USE hmscen
73 ELSE
74 SELECT hmscen
75 ENDIF
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132

```

HMSCEN Screen Layout

```

133 #REGION 1
134 IF WVISIBLE("w_hmscen")
135 ACTIVATE WINDOW w_hmscen SAME
136 ELSE
137 ACTIVATE WINDOW w_hmscen NOSHOW
138 ENDIF
139 @ 1,5 SAY "Scenario:"
140 @ 1,15 GET m.hmscen ;
141 SIZE 1,42 ;
142 DEFAULT " " ;
143 PICTURE "g1h" ;
144 VALID qbk0ingjh()
145 @ 4,5 GET mevent ;
146 PICTURE "g1h" \<New;\<Retrieve;\<Delete;\<Browse;\<Cancel" ;
147 SIZE 1,10,1 ;
148 DEFAULT 1 ;
149 VALID qbk0ingtv()
150
151 IF NOT WVISIBLE("w_hmscen")
152 ACTIVATE WINDOW w_hmscen
153 ENDIF
154
155 READ CYCLE MODAL ;
156 WHEN _qbk0inhbv()
157
158 RELEASE WINDOW w_hmscen
159 SELECT (m.curarea)
160
161 #REGION 0
162 IF m.talkstat = "ON"
163 SET TALK ON
164 ENDIF
165 IF m.compstat = "ON"
166 SET COMPATIBLE ON
167 ENDIF
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

```

HMSCEN Cleanup Code

```

201 #REGION 1
202 SELECT hmscen

```

```

133 USE
134 SELECT (msel)
135 m.ans = m.driven + u + ALLTRIM(STR(m.hmscenid)) + u + m.hmscen
136 RETURN m.ans
137
138 *-----
139 * This procedure will return:
140 * Scenario name (Characters) Scenario id (Numeric)
141 *-----
142 FUNCTION get_hmsc
143 PARAMETER name
144 RELEASE hmscn
145 DIMENSION hmscn[1]
146
147 name = ALLTRIM(name)
148 SELECT DISTINCT hmscen.hmscname, hmscen.hmscid;
149 FROM hmscen;
150 WHERE UPPER(hmscen.hmscname) IN (UPPER(name));
151 INTO ARRAY hmscn
152
153 m.ans=""
154 IF NOT EMPTY(hmscn[1])
155   m.ans=chooser(hmscn,"Select a Material Scenario")
156   IF EMPTY(m.ans)
157     m.index=ASCAN(hmscn,m.ans)
158     m.id = hmscn[m.index + 1]
159     m.ans = ALLTRIM(m.ans) + u + ALLTRIM(STR(m.id))
160   ENDIF
161 RETURN m.ans
162
163 *-----
164 * This procedure show the status of option.
165 * status 1: if new entry allow for add, browse, and cancel
166 * status 2: if entry already exist allow for retrieve, delete, browse
167 =>
168
169
170
171
172 PROCEDURE mevent
173 PARAMETER mnew
174 IF mnew
175   SHOW GET mevent,1 enabled
176   SHOW GET mevent,2 disabled
177   SHOW GET mevent,3 disabled
178 ELSE
179   SHOW GET mevent,1 disabled
180   SHOW GET mevent,2 enabled
181   SHOW GET mevent,3 enabled
182 ENDIF
183 RETURN
184
185
186
187
188
189
190
191
192
193
194
195
196
197

```

_q8k0ingjh	m.hmscen VALID
Function Origin:	
From Screen:	HMSZEN, Record Number: 3
Variable:	m.hmscen
Called By:	VALID Clause
Object Type:	Field
Snippet Number:	1

```

198
199
200 FUNCTION _q8k0ingjh 22 m.hmscen VALID
201 #REGION 1
202 m.hmsc = ""
203 IF NOT EMPTY(m.hmscen)
204   m.hmscen=ALLTRIM(m.hmscen)
205   m.hmscen=IF(m.hmscen="?", "", UPPER(m.hmscen))
206   m.hmsc=get_hmsc(m.hmscen)
207 ENDIF
208 IF EMPTY(m.hmsc)
209   DO mevent WITH .T.
210 ELSE
211   DO mevent WITH .F.
212   m.hmscen = dp(m.hmsc, u, 1)
213   m.hmscenid = VAL(dp(m.hmsc, u, 2))
214 ENDIF
215 SHOW GETS
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231 FUNCTION _q8k0ingtv 22 mevent VALID
232 #REGION 1
233 IF EMPTY(m.hmscen)
234   IF mevent < 4
235     CURSOR = OBJNUM(m.hmscen)
236     =errmsg("Input required",2)
237   SHOW GETS
238   RETURN
239 ENDIF
240
241 DO CASE
242   CASE mevent = 1
243     SET ORDER TO TAG hmscid
244     GO BOTTOM
245     m.hmscenid = hmscid + 1
246     m.driven = "N"
247     CLEAR READ
248   CASE mevent = 2
249     m.driven = "R"
250     CLEAR READ
251   CASE mevent = 3
252     m.driven = "D"
253     SET ORDER TO TAG hmscid
254     GO TOP
255     SEEK m.hmscenid
256     DELETE
257     PACK
258     CLEAR READ
259   CASE mevent = 4
260     m.hmsc = ""
261

```

_q8k0ingtv	mevent VALID
Function Origin:	
From Screen:	HMSZEN, Record Number: 4
Variable:	mevent
Called By:	VALID Clause
Object Type:	Push Button
Snippet Number:	2

22 New

22 Retri

22 Delet

22 Brows

```

261 m.hmsc = get_hmsc(m.hmsc)
262 IF EMPTY(m.hmsc)
263 DO mevent WITH .F.
264 m.hmscen = dp(m.hmsc,u,1)
265 m.hmscenid = VAL(dp(m.hmsc,u,2))
266 ENDIF
267 SHOW GETS
268
269 CASE mevent = 5
270 m.hmscenid = 0
271 m.hmscen = ""
272 m.driven = "C"
273 CLEAR READ
274 ENDCASE
275 RETURN
276
277 *
278 *
279 *
280 *
281 *
282 *
283 *
284 *
285 *
286 *
287 *
288
289 FUNCTION _q8k0inhbv    && Read Level When
290 *
291 * When Code from screen: HMSCEM
292 *
293 #REGION 1
294 DO mevent WITH .T.
295 *: EOF: HMSCEM.ACT

```

_q8k0inhbv	Read Level When
Function Origin:	HMSCEM
From Screen:	READ Statement
Called By:	3
Snippet Number:	

08/26/93	HMSTEP .PRG	14:36:09
Author's Name		
Copyright (c) 1993 Company Name		
Address		
City,	Zip	
Description: This program was automatically generated by GENSCRN.		

PARAMETERS adding, scid, scname

HMSTEP/MS-DOS Setup Code - SECTION 1

```
#REGION 1
m.invalid = .f.
u = ""
m.adding = IIF(PARMETER() = 0, .I., m.adding)
m.gmscid = IIF(PARMETER() < 1, 0, m.scid)
m.gmscname = IIF(PARMETER() < 2, "", m.scname)
m.gchange = IIF(m.adding, .I., .F.)
m.quitflag = .f.
```

```
#REGION 0
REGIONAL m.curarea, m.talkstat, m.compstat
```

```
IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS
m.curarea = SELECT()
```

MS-DOS Window definitions

```
IF NOT EXIST("hmscen") ;
OR UPPER(UTITLE("HMSCEN")) = "HMSCEN.PJX" ;
OR UPPER(UTITLE("HMSCEN")) = "HMSCEN.SCX" ;
OR UPPER(UTITLE("HMSCEN")) = "HMSCEN.MNX" ;
OR UPPER(UTITLE("HMSCEN")) = "HMSCEN.PRG" ;
OR UPPER(UTITLE("HMSCEN")) = "HMSCEN.FRX" ;
```

```
67 OR UPPER(UTITLE("HMSCEN")) = "HMSCEN.OPR"
68 DEFINE WINDOW hmscen ;
69 FROM INT((SROW()-16)/2),INT((SCOL()-78)/2) ;
70 TO INT((SROW()-16)/2)+15,INT((SCOL()-78)/2)+77 ;
71 TITLE "HAZARDOUS MATERIAL SCENARIO" ;
72 FLOAT ;
73 NOCLOSE ;
74 SHADOW ;
75 nonimize ;
76 DOUBLE ;
77 COLOR SCHEME 1
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
```

HMSTEP/MS-DOS Setup Code - SECTION 2

```
#REGION 1
PUSH KEY
*ON KEY LABEL ESC DO EscPressed
m.oldscape = SET("ESCAPE")
m.olddtalk = SET("talk")
m.olddsafe = SET("SAFETY")
m.olddproc = SET("PROC")
SET ESCAPE OFF
SET SAFETY OFF
SET PROCEDURE TO hmsc.prg
m.action = 1
m.step = 0
STORE "" TO m.olddhname, m.olddhmlc, m.olddhmap
STORE "" TO m.hname, m.hmlc, hmap
STORE .f. TO m.validproc
filesucces = openfile("hmstep")
IF filesucces
=errmsg("File doesn't Exist !!!", 2)
DO CANCEL
ENDIF
IF EMPTY(m.gmscid)
SELECT *
FROM hmstep
WHERE hmstep.hmscid = m.gmscid
INTO TABLE hmtemp
ENDIF
SELECT hmtemp
GO TOP
IF m.adding
SCATTER MEMVAR BLANK
m.hmscid = 1
m.hmscid = m.gmscid
m.olddrec = 0
ELSE
SCAN
m.step = m.step + 1
IF hmstep.l = m.step
REPLACE hmstep WITH m.step
ENDIF
ENDSCAN
GO TOP
SCATTER MEMVAR
m.olddrec = RECNO()
```

HMSTEP/MS-DOS Screen Layout

81

HMSTEP/MS-DOS Cleanup Code

```

265 *
266 *
267 *
268
269 #REGION 1
270 POP KEY ALL
271 SET ESCAPE &oldescape
272 SET SAFETY &oldsafe
273 RETURN m.quitflag
274 ***** End of Main Body - Entry Cleanup
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330

```

```

*****
PROCEDURE initvar
*****
IF m.hmcomid>0
  m.hname=get_hmatn(m.hmcomid)
  m.olchname=m.hname
ELSE
  m.hname=SPACE(46)
  m.olchname=SPACE(46)
ENDIF

IF m.hmlcid>0
  m.hmlc=get_hmlcn(m.hmlcid)
  m.olchmlc=m.hmlc
ELSE
  m.hmlc=SPACE(46)
  m.olchmlc=SPACE(46)
ENDIF

IF m.hmwpid>0
  m.hmwp=get_hmwpn(m.hmwpid)
  m.olchmwp=m.hmwp
ELSE
  m.hmwp=SPACE(80)
  m.olchmwp=SPACE(80)
ENDIF
RETURN
*****
PROCEDURE adoption
*****
SHOW GET m.action disabled
SHOW GET m.save enabled
SHOW GET m.hname enabled
SHOW GET m.hmlc enabled
SHOW GET m.hmwp enabled
SHOW GET m.pernum enabled
SHOW GET m.durnum enabled
SHOW GET m.qtytnum enabled
SHOW GET m.unit enabled
RETURN
*****
PROCEDURE CHANGE

```

```

*****
m.oldexact = SET( "EXACT" )
SET EXACT ON
m.change =(hmtmp.hmcomid <> m.hmcomid;
OR hmtmp.hmlcid <> m.hmlcid;
OR hmtmp.hmwpid <> m.hmwpid;
OR hmtmp.durnum <> m.durnum;
OR hmtmp.qtytnum <> m.qtytnum;
OR hmtmp.pernum <> m.pernum;
OR UPPER(ALLTRIM(hmtmp.unit)) <> UPPER(ALLTRIM(m.unit)))
SET EXACT &oldexact
RETURN m.change
*****
FUNCTION get_hmat
*****
PARAMETER name
RELEASE hmt
DIMENSION hmt[1]
hmt[1]=" "
m.oldfile=SELECT()

SELECT hmcom.common, hmcom.hmcomid;
FROM hmcom;
WHERE hmcom.common IN (ALLTRIM(m.name));
INTO ARRAY hmt

m.ans=""
IF NOT EMPTY(hmt[1])
  m.ans=chooser(hmt,"Select a Material")
  IF EMPTY(m.ans)
    m.index = ASCAN(hmt,m.ans)
    m.id = hmt[m.index + 1]
    m.ans = m.ans + u + STR(m.id)
  ENDIF
ELSE
  =errmsg(m.name + " was not found",1)
  m.ans=""
ENDIF

SELECT (m.oldfile)
RETURN m.ans
*****
FUNCTION get_hmatn
*****
PARAMETER m.id

SELECT hmcom.common;
FROM hmcom;
WHERE hmcom.hmcomid = m.id;
INTO ARRAY X

m.name=X[1]
RETURN m.name
*****
*GET LIFE CYCLE PHASE
FUNCTION get_hmlc
*****
PARAMETER m.hmlc

RELEASE hmt
DIMENSION hmt[1]
hmt[1]=" "

```

```

397 m.oldfile=SELECT()
398
399 SELECT hmlc.hmlc, hmlc.hmlcid;
400 FROM hmlc;
401 WHERE hmlc.hmlc IN (ALLTRIM(m.hmlc));
402 ORDER BY hmlcid;
403 INTO ARRAY hmn
404
405 m.ans=""
406 IF NOT EMPTY(hmn[1])
407   m.ans=chooser(ahmn,"Select a Life Cycle Phase")
408   IF EMPTY(m.ans)
409     m.index = ASCAN(hmn,m.ans)
410     m.id = hmn[m.index + 1]
411     m.ans = m.ans + u + STR(m.id)
412   ENDIF
413 ELSE
414   =errmsg(m.hmlc + " was not found",1)
415 ENDIF
416
417 SELECT (m.oldfile)
418 RETURN m.ans
419
420 *****
421 FUNCTION get hmlcn
422 *****
423 PARAMETER m.id
424
425 SELECT hmlc.hmlc;
426 FROM hmlc;
427 WHERE hmlc.hmlcid = m.id;
428 INTO ARRAY X
429
430 m.name=X[1]
431 RETURN m.name
432
433 *****
434 *Working Processes
435 FUNCTION get hmw
436 *****
437 PARAMETER m.match
438 RELEASE hmn
439 DIMENSION hmn[1]
440 hmn[1]="
441 m.oldfile=SELECT()
442 m.match = UPPER(ALLTRIM(m.match))
443 SELECT hmw.hmw, hmw.hmwpid;
444 FROM hmw;
445 WHERE UPPER(hmw.hmw) IN (ALLTRIM(m.match));
446 INTO ARRAY hmn
447
448 m.ans=""
449 IF NOT EMPTY(hmn[1])
450   m.ans=chooser(ahmn,"Select a Process")
451   IF EMPTY(m.ans)
452     m.index = ASCAN(hmn,m.ans)
453     m.id = hmn[m.index + 1]
454     m.ans = m.ans + u + STR(m.id)
455   ENDIF
456 ELSE
457   =errmsg(m.hmw + " was not found",1)
458 ENDIF
459
460 SELECT (m.oldfile)
461 RETURN m.ans
462

```

```

463 *****
464 FUNCTION get hmwpid
465 *****
466 PARAMETER m.id
467
468 SELECT hmw.hmw;
469 FROM hmw;
470 WHERE hmw.hmwpid = m.id;
471 INTO ARRAY X
472
473 m.name=X[1]
474 RETURN m.name
475
476 *****
477 PROCEDURE get hmunid
478 *****
479
480 PRIVATE m.unit
481 m.unit = ""
482 IF EMPTY(m.hmcomid) AND EMPTY(m.hmlcid) AND EMPTY(m.hmwpid)
483   DIMENSION X[1]
484   X[1]=""
485   SELECT hmtab.tabid, hmtab.hmunid;
486   FROM hmtab;
487   WHERE hmtab.hmcomid = m.hmcomid;
488   AND hmtab.hmlcid = m.hmlcid;
489   AND hmtab.hmwpid = m.hmwpid;
490   INTO ARRAY X
491
492 IF EMPTY(X[1])
493   FOR i = 1 TO ALEN(X) STEP 2
494     m.unit = X[i+1]
495     IF EMPTY(m.unit)
496       EXIT
497     ENDIF
498   ENDFOR
499   m.validproc = .T.
500   SHOW GETS
501 ELSE
502   =errmsg("This Material Processing isn't defined in the table")
503   m.validproc = .F.
504 ENDIF
505
506 RETURN m.unit
507
508 *****
509 PROCEDURE scsave
510 *****
511 msel = SELECT()
512 SELECT hmwstep
513 IF schange
514   SET ORDER TO TAG hmscid
515   DELETE FOR hmscid = m.ghmscid
516   PACK
517   APPEND FROM hmwtemp
518   USE hmscen
519   SET ORDER TO TAG hmscid
520   IF !SEEK(m.ghmscid)
521     APPEND BLANK
522     REPLACE hmscid WITH m.ghmscid
523     REPLACE hmscname WITH m.ghmscname
524   ENDIF
525 ENDIF
526 *****
527 FUNCTION datacheck
528 *****

```

```

529 DO CASE
530 CASE EMPTY(m.hmcomid)
531 = errmsg("Missing Material",1)
532 CUROBJ = OBJNUM(hmname)
533 Check = .F.
534 CASE EMPTY(m.hmlcid)
535 = errmsg("Missing Life Cycle Phase",1)
536 CUROBJ = OBJNUM(hmlc)
537 Check = .F.
538 CASE EMPTY(m.hmpid)
539 = errmsg("Missing Working Process",1)
540 CUROBJ = OBJNUM(hmwp)
541 Check = .F.
542 CASE EMPTY(m.pernum) AND EMPTY(m.durnum) AND EMPTY(m.qtnum)
543 = errmsg("Enter either number of People, Duration, or Quantity",1)
544 CUROBJ = OBJNUM(pernum)
545 Check = .F.
546 OTHERWISE
547 =get hmunit()
548 IF !m.validproc
549 CUROBJ = OBJNUM(hmname)
550 Check = .F.
551 ELSE
552 check = .T.
553 ENDIF
554 ENDCASE
555 RETURN check
556
557 *
558 *
559 *
560 *
561 *
562 *
563 *
564 *
565 *
566 *
567 *
568 *
569 *
570 *
571 FUNCTION qipovascn && m.Action VALID
572 #REGION 1-
573 m.prevhmcomid = m.hmcomid
574 m.prevhmlcid = m.hmlcid
575 m.oldrec = IIF(RECCOUNT() = 0, 0, RECNO())
576 DO CASE
577 CASE m.action = 1
578 msanephase = yesno("Same Phase?", "Yes", "No")
579 SCATTER MENVAR BLANK
580 m.hmstep = RECCOUNT() + 1
581 IF msanephase
582 m.hmcomid = m.prevhmcomid
583 m.hmlcid = m.prevhmlcid
584 CUROBJ = OBJNUM(m.hmwp)
585 ELSE
586 CUROBJ = OBJNUM(m.hmname)
587 ENDIF
588 DO initvar
589 SHOW GETS
590 DO adoption
591 m.adding = .T.
592 CASE m.action = 2
593 validproc = .T.
594

```

_QIPOVASCN	m.Action VALID
Function Origin:	
From Platform:	MS-DOS
From Screen:	HMSTEP,
Variable:	m.Action
Called By:	VALID Clause
Snippet Number:	1

```

595 DO adoption
596 CASE m.action = 3
597 SKIP
598 IF EOF()
599 ?? CHR( 7 )
600 WAIT "Last step" WINDOW NOWAIT
601 GO BOTTOM
602 ENDIF
603
604 CASE m.action = 4
605 SKIP -1
606 IF BOF()
607 ?? CHR( 7 )
608 WAIT "First step" WINDOW NOWAIT
609 GO TOP
610 ENDIF
611
612 CASE m.action = 5
613 IF m.gchange
614 m.scsave = yesno("Save Scenario?", "YES", "NO")
615 IF m.scsave
616 DO scsave
617 ENDIF
618 ENDIF
619 CLEAR READ
620 ENDCASE
621
622 IF m.action > 2 AND m.action < 5
623 m.oldrec = RECNO()
624 SCATTER MENVAR
625 DO initvar
626 SHOW GETS disabled
627 SHOW GET action enabled
628 ENDIF
629
630 *
631 *
632 *
633 *
634 *
635 *
636 *
637 *
638 *
639 *
640 *
641 *
642 *
643 *
644 *
645 *
646 *
647 *
648 #REGION 1-
649 m.oldhmname = m.hmname
650 FUNCTION qipovasvs && m.hmname WHEN
651
652 *
653 *
654 *
655 *
656 *
657 *
658 *
659 *
660

```

_QIPOVASVS	m.hmname WHEN
Function Origin:	
From Platform:	MS-DOS
From Screen:	HMSTEP,
Variable:	m.hmname
Called By:	WHEN Clause
Snippet Number:	2

_QIPOVASYS	m.hmname VALID
Function Origin:	
From Platform:	MS-DOS
From Screen:	HMSTEP,
Variable:	m.hmname


```

661 * Called By: VALID Clause
662 * Snippet Number: 3
663 *
664 *
665 *
666 * FUNCTION _qip0vasya && m.hname VALID
667 #REGION 1
668 IF m.olchname <> m.hname
669 IF NOT EMPTY(m.hname)
670 m.hname=ALLTRIM(m.hname)
671 m.hname=IF(m.hname="?", "", UPPER(m.hname))
672 m.hdata=ALLTRIM(get_hmat(m.hname))
673 IF NOT EMPTY(m.hdata)
674 m.hname = dp(m.hdata,u,1)
675 m.hcomid = VAL(dp(m.hdata,u,2))
676 ELSE
677 m.hname = SPACE(46)
678 ENDIF
679 SHOW GETS
680 m.t1 = get_hmunit()
681 IF EMPTY(m.t1)
682 m.unit = m.t1
683 ENDIF
684 ENDIF
685 SHOW GETS
686 ENDIF
687
688 *
689 *
690 *
691 *
692 *
693 *
694 *
695 *
696 *
697 *
698 *
699 *
700 *
701 *
702 *
703 *
704 *
705 *
706 *
707 *
708 *
709 *
710 *
711 *
712 *
713 *
714 *
715 *
716 *
717 *
718 *
719 *
720 *
721 *
722 *
723 *
724 *
725 *
726 *

```

```

_QIPOVAT7G      m.hmlc WHEN
Function Origin:
From Platform:  MS-DOS
From Screen:    HMSTEP,
Variable:       m.hmlc
Called By:      WHEN Clause
Snippet Number: 4
Record Number: 19

```

```

FUNCTION _qip0vat7g && m.hmlc WHEN
#REGION 1
m.olchmlc=m.hmlc

```

```

_QIPOVAT9X      m.hmlc VALID
Function Origin:
From Platform:  MS-DOS
From Screen:    HMSTEP,
Variable:       m.hmlc
Called By:      VALID Clause
Snippet Number: 5
Record Number: 19

```

```

FUNCTION _qip0vat9x && m.hmlc VALID
#REGION 1
IF m.olchmlc <> m.hmlc
m.hmlc=ALLTRIM(m.hmlc)
m.hmlc=IF(m.hmlc="?", "", UPPER(m.hmlc))

```

```

727 * m.hdata=get_hmlc(m.hmlc)
728 * IF NOT EMPTY(m.hdata)
729 * m.hmlc = dp(m.hdata,u,1)
730 * n.hmlcid = VAL(dp(m.hdata,u,2))
731 * ELSE
732 * m.hmlc = SPACE(46)
733 * ENDIF
734 * SHOW GETS
735 * m.t1 = get_hmunit()
736 * IF EMPTY(m.t1)
737 * m.unit = m.t1
738 * ENDIF
739 * ENDIF
740 * SHOW GETS
741 * ENDIF
742 *
743 *
744 *
745 *
746 *
747 *
748 *
749 *
750 *
751 *
752 *
753 *
754 *
755 *
756 *
757 *
758 *
759 *
760 *
761 *
762 *
763 *
764 *
765 *
766 *
767 *
768 *
769 *
770 *
771 *
772 *
773 *
774 *
775 *
776 *
777 *
778 *
779 *
780 *
781 *
782 *
783 *
784 *
785 *
786 *
787 *
788 *
789 *
790 *
791 *
792 *

```

```

_QIPOVATIU      m.hmap WHEN
Function Origin:
From Platform:  MS-DOS
From Screen:    HMSTEP,
Variable:       m.hmap
Called By:      WHEN Clause
Snippet Number: 6
Record Number: 21

```

```

FUNCTION _qip0vatiu && m.hmap WHEN
#REGION 1
m.olchmap=m.hmap

```

```

_QIPOVATLC      m.hmap VALID
Function Origin:
From Platform:  MS-DOS
From Screen:    HMSTEP,
Variable:       m.hmap
Called By:      VALID Clause
Snippet Number: 7
Record Number: 21

```

```

FUNCTION _qip0vatlc && m.hmap VALID
#REGION 1
IF m.olchmap <> m.hmap
m.hmap=ALLTRIM(m.hmap)
m.hmap=IF(m.hmap="?", "", UPPER(m.hmap))
m.hdata=ALLTRIM(get_hmap(m.hmap))
IF NOT EMPTY(m.hdata)
m.hmap = dp(m.hdata,u,1)
m.hmapid = VAL(dp(m.hdata,u,2))
ELSE
m.hmap = SPACE(80)
ENDIF
SHOW GETS
m.t1 = get_hmunit()
IF EMPTY(m.t1)
m.unit = m.t1
ENDIF

```

```
793 _ENDIF
794 SHOW GETS
795 _ENDIF
796
797 *
798 *
799 *
800 *
801 *
802 *
803 *
804 *
805 *
806 *
807 *
808 *
809 *
810 *
811 *
812 *
813 *
814 *
815 *
816 *
817 *
818 *
819 *
820 *
821 *
822 *
823 *
824 *
825 *
826 *
827 *
828 *
829 *
830 *
831 *
832 *
833 *
834 *
835 *
836 *
837 *
838 *
839 *
840 *
841 *
842 *
843 *
844 *
845 *
846 *
847 *
848 *
849 *
850 *
851 *
852 *
853 *
854 *
855 *
856 *
857 *
858 *
```

_QIP0VATWZ m.Save VALID

Function Origin:

From Platform: MS-DOS

From Screen: HMSTEP,

Variable: m.Save, Record Number: 26

Called By: VALID Clause

Snippet Number: 8

```
FUNCTION _qip0vatwz      && m.Save VALID
#REGION 1
DO CASE
CASE m.save = 1 && Selected Save Button
m.check = datacheck()
IF m.check
m.hmscid = m.ghmscid
IF m.adding && Adding a new record
APPEND BLANK
GATHER MEMVAR
m.gchange = .T.
ELSE
DO CHANGE
IF m.change && Changing an old record
GATHER MEMVAR
m.gchange = .T.
ENDIF
ENDIF
ELSE
m.oldrec = RECNO()
RETURN
ENDIF
CASE m.save = 2
IF RECCOUNT() = 0
QUIT = .F.
IF EMPTY(m.hname) AND EMPTY(m.hmlc) AND EMPTY(hmwp)
QUIT = yesno("Do you want to", "Quit", "Continue")
ENDIF
IF QUIT
m.quitflag = .T.
CLEAR READ
ELSE
SCATTER MEMVAR BLANK
DO initvar
SHOW GETS
DO adoption
m.adding = .T.
m.change = .F.
m.quitflag = .F.
RETURN
ENDIF
ENDIF
ENDCASE
IF EMPTY(m.oldrec)
GO m.oldrec
SCATTER MEMVAR
ENDIF
```

```
859 DO initvar
860 SHOW GETS disabled
861 SHOW GET action enabled
862
863 m.adding = .F.
864 m.change = .F.
865
866 *
867 *
868 *
869 *
870 *
871 *
872 *
873 *
874 *
875 *
876 *
877 *
878 *
879 *
880 *
881 *
882 *
883 *
884 *
885 *
886 *
887 *
888 *
889 *
890 *
891 *
892 *
```

_QIP0VAUD2 Read Level When

Function Origin:

From Platform: MS-DOS

From Screen: HMSTEP

Called By: READ Statement

Snippet Number: 9

```
FUNCTION _qip0vaud2      && Read Level When
* When Code from screen: HMSTEP
*
#REGION 1
IF m.adding
m.action = 1
SHOW GETS DISABLE
SHOW GET action enabled
ENDIF
*: EOF: HMSTEP.ACT
```

```

1  *:*:*****
=> *****
2  *:*:
3  *:*: Procedure file: C:\HAZMAT\GHW\WORK\HMZTAB.PRG
4  *:*:
5  *:*: System: Hazardous Material Life-Cycle Cost Model
6  *:*: Author: Naval Health Research Center
7  *:*: Copyright (c) 1993, Naval Health Research Center
8  *:*: Last modified: 09/10/93 8:12
9  *:*:
10 *:*: Set by: HMENU.MPR
11 *:*:
12 *:*: Uses: TLV.DBF
13 *:*:
14 *:*: CDX files: TLV.CDX
15 *:*:
16 *:*: Documented 12/01/93 at 11:32 ***** FoxDoc version 2.10f
17 *:*:*****
=> *****
18 * hmtab-prg gp 8/9/93
19 * OSHA 2 table look up from tlv table from Smallett
20 USE tlv
21 BROWSE FIELDS substance, cas_no:H='CAS Number',pel_twa:H='TWA (mg/m3)
=> ",,;
22 USE
23 pel_stel:H='STEL (mg/m3)' NOMODIFY NORMAL
24 *:* EOF: HMZTAB.ACT

```

```

* * * * *
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66
* * * * *
08/28/92      MEMOEDIT.PRQ      10:02:45
* * * * *
Hoa L. Ly
* * * * *
Copyright (c) 1992 Naval Health Research Center
P..Box 85122
San Diego, CA 92186-5122
* * * * *
Description:
This program was automatically generated by GENSCRN.
* * * * *

* * * * *
MEMOEDIT Setup Code - SECTION 1
* * * * *

* * * * *
#REGION 1
PARAMETER TEXT, TITLE, editallow
PRIVATE ALL
DO CASE
  -CASE PARAMETER() = 0
    m.text = ""
    m.title = ""
    m.editallow = .T.
  -CASE PARAMETER() = 1
    m.title = ""
    m.editallow = .T.
  -CASE PARAMETER() = 2
    m.editallow = .T.
  -ENDCASE
* * * * *

* * * * *
#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat
* * * * *
IF SET("TALK") = "ON"
  SET TALK OFF
  m.talkstat = "ON"
ELSE
  m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS
m.currarea = SELECT()
* * * * *

* * * * *
Window definitions
* * * * *

* * * * *
IF NOT VEXIST("memoedit")
DEFINE WINDOW memoedit ;
FROM INT((SROW()-20)/2), INT((SCOL()-77)/2) ;
TO INT((SROW()-20)/2)+19, INT((SCOL()-77)/2)+76 ;
* * * * *

```

133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178

```

RELEASE WINDOW memoeedit
SELECT (m.curraea)

#REGION 0
IF m.talkstat = "ON"
  SET TALK ON
ENDIF
IF m.compstat = "ON"
  SET COMPATIBLE ON
ENDIF
  
```

MEMOEDIT Cleanup Code

```

#REGION 1
RETURN m.text
  
```

_Q8L0LJ6F0 m.save VALID

Function Origin: MEMOEDIT, Record Number: 4

From Screen: m.save
Variable: m.save
Called By: VALID Clause
Object Type: Push Button
Snippet Number: 1

```

FUNCTION _q8l0lj6f0 && m.save VALID
#REGION 1
IF m.save = 2
  m.text = m.oldtext
ENDIF
  
```

*: EOF: MEMOEDIT.ACT

```

1  * *****
2  *
3  * Procedure file: C:\HAZMAT\GHW\WORK\MEMOWIN.PRG
4  *
5  * System: Hazardous Material Life-Cycle Cost Model
6  * Author: Naval Health Research Center
7  * Copyright (c) 1993, Naval Health Research Center
8  * Last modified: 08/17/93 8:54
9  *
10 * Procs & Fncts: _QIG01XOXD()
11 * : _QIG01XPJO()
12 *
13 * Set by: HWVIEW() (function in HWREF.PRG)
14 *
15 * Calls: _QIG01XOXD() (function in MEMOWIN.PRG)
16 * : _QIG01XPJO() (function in MEMOWIN.PRG)
17 *
18 * Documented 12/01/93 at 11:32 FoxDoc version 2.10f
19 * *****
20 *
21 * *****
22 * * 08/17/93 MEMOWIN.SPR 08:50:04
23 *
24 * *****
25 * * G. Pang
26 *
27 * * Copyright (c) 1993 Company Name
28 * * Address
29 * * City, Zip
30 *
31 * * Description:
32 * * This program was automatically generated by GENSCRN.
33 *
34 * *****
35 *
36 *
37 * PARAMETERS TEXT, TITLE
38 *
39 * DO CASE
40 * CASE _dos
41 *
42 * *****
43 * *
44 * * MEMOWIN/MS-DOS Setup Code - SECTION 1
45 * *
46 * *****
47 *
48 * #REGION 1
49 * * PRIVATE all
50 *
51 * DO CASE
52 * CLEAR
53 * CASE PARAMETER() = 0
54 * m.title = ""
55 * m.text = ""
56 * CASE PARAMETER() = 1
57 * m.title = ""
58 * ENDCASE
59 *
60 * #REGION 0
61 * REGIONAL m.curarea, m.talkstat, m.compstat
62 *
63 *
64 *

```

```

65 IF SET("TALK") = "ON"
66 SET TALK OFF
67 m.talkstat = "ON"
68
69 ELSE
70 m.talkstat = "OFF"
71
72 ENDIF
73 m.compstat = SET("COMPATIBLE")
74 SET COMPATIBLE FOXPLUS
75
76 m.curarea = SELECT()
77
78 * *****
79 * *
80 * * MS-DOS Window definitions
81 * *
82 * *****
83 *
84 * IF NOT WEXIST("memowin") ;
85 * OR UPPER(UTITLE("MEMOWIN")) == "MEMOWIN.PJK" ;
86 * OR UPPER(UTITLE("MEMOWIN")) == "MEMOWIN.SCK" ;
87 * OR UPPER(UTITLE("MEMOWIN")) == "MEMOWIN.MHX" ;
88 * OR UPPER(UTITLE("MEMOWIN")) == "MEMOWIN.PRG" ;
89 * OR UPPER(UTITLE("MEMOWIN")) == "MEMOWIN.FBX" ;
90 * OR UPPER(UTITLE("MEMOWIN")) == "MEMOWIN.QPR" ;
91 *
92 * DEFINE WINDOW memowin ;
93 * FROM INT((SROW()-26)/2), INT((SCOL()-80)/2) ;
94 * TO INT((SROW()-26)/2)+25, INT((SCOL()-80)/2)+79 ;
95 * NOFLOAT ;
96 * NOCLOSE ;
97 * SHADOW ;
98 * NOMINIMIZE ;
99 * NONE ;
100 * COLOR, SCHEME 1
101
102 ENDIF
103
104 * *****
105 * * MEMOWIN/MS-DOS Screen Layout
106 * *
107 * *****
108 *
109 * #REGION 1
110 * IF WISIBLE("memowin")
111 * ACTIVATE WINDOW memowin SAME
112 * ELSE
113 * ACTIVATE WINDOW memowin NOSHOW
114 * ENDIF
115
116 @ 1,2 GET m.title ;
117 SIZE 1,74 ;
118 DEFAULT " " ;
119 PICTURE "a1" ;
120 DISABLE
121 @ 2,2 EDIT m.text ;
122 SIZE 20,77,0 ;
123 DEFAULT " " ;
124 SCROLL ;
125 NOMODIFY
126
127 @ 23,26 GET m.save ;
128 PICTURE "a" HT \<Print;\<Another;\<Quit" ;
129 SIZE 1,9,1 ;
130 DEFAULT 1 ;
131 VALID _qig01xoxd()

```

91

```

263 READ CYCLE
264
265 RELEASE WINDOW memowin
266 SELECT (m.curarea)
267
268
269 #REGION 0
270 IF m.talkstat = "ON"
271 SET TALK ON
272 ENDIF
273 IF m.compstat = "ON"
274 SET COMPATIBLE ON
275 ENDIF
276
277
278 *****
279 *
280 *
281 *
282 *
283 *
284
285 #REGION 1
286 RETURN
287
288
289
290
291 *****
292 *
293 *
294 *
295 *
296 *
297 *
298 *
299 *
300 *
301 *
302 *
303 *
304 *
305 *****
306
307
308 *****
309 *
310 *
311 *
312 *
313 *
314 *
315 *
316 *
317 *
318 *
319 *
320 *
321 *
322 *
323 *****
324
325
326
327
328 *****
329 *
330 *
331 *
332 *
333 *
334 *
335 *
336 *
337 *
338 *
339 *
340 *
341 *
342 *
343 *****
344
345
346
347
348 *****
349 *
350 *
351 *
352 *
353 *
354 *
355 *
356 *
357 *
358 *
359 *
360 *
361 *
362 *
363 *****
364
365
366
367
368 *****
369 *
370 *
371 *
372 *
373 *
374 *
375 *
376 *
377 *
378 *
379 *
380 *
381 *
382 *
383 *****
384
385
386
387
388 *****
389 *
390 *
391 *
392 *
393 *
394 *
395 *
396 *
397 *
398 *
399 *
400 *
401 *
402 *
403 *****
404
405
406
407
408 *****
409 *
410 *
411 *
412 *
413 *
414 *
415 *
416 *
417 *
418 *
419 *
420 *
421 *
422 *
423 *****
424
425
426
427
428 *****
429 *
430 *
431 *
432 *
433 *
434 *
435 *
436 *
437 *
438 *
439 *
440 *
441 *
442 *
443 *****
444
445
446
447
448 *****
449 *
450 *
451 *
452 *
453 *
454 *
455 *
456 *
457 *
458 *
459 *
460 *
461 *
462 *
463 *****
464
465
466
467
468 *****
469 *
470 *
471 *
472 *
473 *
474 *
475 *
476 *
477 *
478 *
479 *
480 *
481 *
482 *
483 *****
484
485
486
487
488 *****
489 *
490 *
491 *
492 *
493 *
494 *
495 *
496 *
497 *
498 *
499 *
500 *
501 *
502 *
503 *****
504
505
506
507
508 *****
509 *
510 *
511 *
512 *
513 *
514 *
515 *
516 *
517 *
518 *
519 *
520 *
521 *
522 *
523 *****
524
525
526
527
528 *****
529 *
530 *
531 *
532 *
533 *
534 *
535 *
536 *
537 *
538 *
539 *
540 *
541 *
542 *
543 *****
544
545
546
547
548 *****
549 *
550 *
551 *
552 *
553 *
554 *
555 *
556 *
557 *
558 *
559 *
560 *
561 *
562 *
563 *****
564
565
566
567
568 *****
569 *
570 *
571 *
572 *
573 *
574 *
575 *
576 *
577 *
578 *
579 *
580 *
581 *
582 *
583 *****
584
585
586
587
588 *****
589 *
590 *
591 *
592 *
593 *
594 *
595 *
596 *
597 *
598 *
599 *
600 *
601 *
602 *
603 *****
604
605
606
607
608 *****
609 *
610 *
611 *
612 *
613 *
614 *
615 *
616 *
617 *
618 *
619 *
620 *
621 *
622 *
623 *****
624
625
626
627
628 *****
629 *
630 *
631 *
632 *
633 *
634 *
635 *
636 *
637 *
638 *
639 *
640 *
641 *
642 *
643 *****
644
645
646
647
648 *****
649 *
650 *
651 *
652 *
653 *
654 *
655 *
656 *
657 *
658 *
659 *
660 *
661 *
662 *
663 *****
664
665
666
667
668 *****
669 *
670 *
671 *
672 *
673 *
674 *
675 *
676 *
677 *
678 *
679 *
680 *
681 *
682 *
683 *****
684
685
686
687
688 *****
689 *
690 *
691 *
692 *
693 *
694 *
695 *
696 *
697 *
698 *
699 *
700 *
701 *
702 *
703 *****
704
705
706
707
708 *****
709 *
710 *
711 *
712 *
713 *
714 *
715 *
716 *
717 *
718 *
719 *
720 *
721 *
722 *
723 *****
724
725
726
727
728 *****
729 *
730 *
731 *
732 *
733 *
734 *
735 *
736 *
737 *
738 *
739 *
740 *
741 *
742 *
743 *****
744
745
746
747
748 *****
749 *
750 *
751 *
752 *
753 *
754 *
755 *
756 *
757 *
758 *
759 *
760 *
761 *
762 *
763 *****
764
765
766
767
768 *****
769 *
770 *
771 *
772 *
773 *
774 *
775 *
776 *
777 *
778 *
779 *
780 *
781 *
782 *
783 *****
784
785
786
787
788 *****
789 *
790 *
791 *
792 *
793 *
794 *
795 *
796 *
797 *
798 *
799 *
800 *
801 *
802 *
803 *****
804
805
806
807
808 *****
809 *
810 *
811 *
812 *
813 *
814 *
815 *
816 *
817 *
818 *
819 *
820 *
821 *
822 *
823 *****
824
825
826
827
828 *****
829 *
830 *
831 *
832 *
833 *
834 *
835 *
836 *
837 *
838 *
839 *
840 *
841 *
842 *
843 *****
844
845
846
847
848 *****
849 *
850 *
851 *
852 *
853 *
854 *
855 *
856 *
857 *
858 *
859 *
860 *
861 *
862 *
863 *****
864
865
866
867
868 *****
869 *
870 *
871 *
872 *
873 *
874 *
875 *
876 *
877 *
878 *
879 *
880 *
881 *
882 *
883 *****
884
885
886
887
888 *****
889 *
890 *
891 *
892 *
893 *
894 *
895 *
896 *
897 *
898 *
899 *
900 *
901 *
902 *
903 *****
904
905
906
907
908 *****
909 *
910 *
911 *
912 *
913 *
914 *
915 *
916 *
917 *
918 *
919 *
920 *
921 *
922 *
923 *****
924
925
926
927
928 *****
929 *
930 *
931 *
932 *
933 *
934 *
935 *
936 *
937 *
938 *
939 *
940 *
941 *
942 *
943 *****
944
945
946
947
948 *****
949 *
950 *
951 *
952 *
953 *
954 *
955 *
956 *
957 *
958 *
959 *
960 *
961 *
962 *
963 *****
964
965
966
967
968 *****
969 *
970 *
971 *
972 *
973 *
974 *
975 *
976 *
977 *
978 *
979 *
980 *
981 *
982 *
983 *****
984
985
986
987
988 *****
989 *
990 *
991 *
992 *
993 *
994 *
995 *
996 *
997 *
998 *
999 *
1000 *
1001 *
1002 *
1003 *****
1004
1005
1006
1007
1008 *****
1009 *
1010 *
1011 *
1012 *
1013 *
1014 *
1015 *
1016 *
1017 *
1018 *
1019 *
1020 *
1021 *
1022 *
1023 *****
1024
1025
1026
1027
1028 *****
1029 *
1030 *
1031 *
1032 *
1033 *
1034 *
1035 *
1036 *
1037 *
1038 *
1039 *
1040 *
1041 *
1042 *
1043 *****
1044
1045
1046
1047
1048 *****
1049 *
1050 *
1051 *
1052 *
1053 *
1054 *
1055 *
1056 *
1057 *
1058 *
1059 *
1060 *
1061 *
1062 *
1063 *****
1064
1065
1066
1067
1068 *****
1069 *
1070 *
1071 *
1072 *
1073 *
1074 *
1075 *
1076 *
1077 *
1078 *
1079 *
1080 *
1081 *
1082 *
1083 *****
1084
1085
1086
1087
1088 *****
1
```



```

1  * *****
2  *
3  * Procedure file: C:\HAZMAT\GHM\WORK\YESNO.PRG
4  *
5  * System: Hazardous Material Life-Cycle Cost Model
6  * Author: Naval Health Research Center
7  * Copyright (c) 1993, Naval Health Research Center
8  * Last modified: 09/10/93 8:12
9  *
10 * Set by: HMSC.PRG
11 * : _QIPOVASCN() (function in HMSTEP.PRG)
12 * : _QIPOVATMZ() (function in HMSTEP.PRG)
13 *
14 * Calls: ERRMSG.PRG
15 *
16 * Documented 12/01/93 at 11:32 FoxDoc version 2.10f
17 * *****
18 *
19 * YESNO.PRG 22:21:11 *
20 *
21 * *****
22 *
23 * * Description:
24 * * This program was automatically generated by GENSCRN.
25 * *
26 * *****
27 *
28 *
29 * *****
30 *
31 * YESNO Setup Code - SECTION 1
32 *
33 * *****
34 *
35 *
36 *
37 *
38 * #REGION 1
39 *
40 * PARAMETERS MESSAGE, ok, CANCEL
41 * PRIVATE ALL
42 *
43 * ON ERROR DO errmsg WITH MESSAGE()
44 *
45 * * If no message is sent then create message
46 *
47 * IF PARAMETERS()=0
48 * m.message="Is this correct?"
49 * ENDIF
50 *
51 * * Ok and Cancel are used for prompts in the push buttons
52 * IF PARAMETERS() < 3
53 * * Default to normal prompts
54 * m.ok = "OK"
55 * m.cancel = "Cancel"
56 * ENDIF
57 *
58 * * Truncate any message longer than 50 characters
59 * MESSAGE=IF( TYPE("m.message")="C", "", m.message)
60 * IF LEN( m.message ) > 50
61 * * The message is centered in an ASAY
62 * m.message = SUBSTR( m.message, 1, 50 )
63 * ENDIF
64

```

```

65 m.message=UPPER(m.message)
66 m.ok=UPPER(m.ok)
67 m.cancel=UPPER(m.cancel)
68
69 PUSH KEY CLEAR
70
71 #REGION 0
72 REGIONAL m.curarea, m.talkstat, m.compstat
73
74 m.talkstat=SET("TALK")
75 SET TALK OFF
76
77 m.compstat = SET("COMPATIBLE")
78 SET COMPATIBLE FOXPLUS
79
80 * *****
81 * * Window definitions
82 * *
83 * *****
84 *
85 *
86 *
87 * IF NOT MEXIST("yesno")
88 * DEFINE WINDOW yesno ;
89 * FROM INT((SROW()-7)/2),INT((SCOL()-48)/2) ;
90 * TO INT((SROW()-7)/2)+6,INT((SCOL()-48)/2)+47 ;
91 * FLOAT ;
92 * NOCLOSE ;
93 * SHADOW ;
94 * DOUBLE ;
95 * COLOR SCHEME 5
96 * ENDIF
97
98 * *****
99 * * YESNO Screen Layout
100 * *
101 * *
102 * *
103 * *
104 *
105 * #REGION 1
106 * IF WISIBLE("yesno")
107 * ACTIVATE WINDOW yesno SAME
108 * ELSE
109 * ACTIVATE WINDOW yesno NOSHOWN
110 * ENDIF
111 * @ 1,2 SAY PADC( m.message, 40 ) ;
112 * SIZE 1,41
113 * @ 3,8 GET m.answer ;
114 * PICTURE "a=HT \i<&OK;\<&Cancel" ;
115 * SIZE 1,12,5 ;
116 * DEFAULT 1
117 *
118 * IF NOT WISIBLE("yesno")
119 * ACTIVATE WINDOW yesno
120 * ENDIF
121
122 READ CYCLE MODAL
123
124 RELEASE WINDOW yesno
125
126 #REGION 0
127
128 IF m.talkstat = "ON"
129 SET TALK ON
130

```

```

131 ELSE SET TALK OFF
132 SET TALK OFF
133 ENDIF
134
135 IF m.compstat = "ON"
136 SET COMPATIBLE ON
137 ENDIF
138
139 * *****
140 * *
141 * *
142 * * YESNO Cleanup Code *
143 * *
144 * *
145 * *
146 *
147 #REGION 1
148 POP KEY
149
150 * Convert the numeric value of m.Answer from 1|2 to .T.|.F.
151 * If the user selected OK and didn't exit with Escape
152 IF m.answer = 1 AND LASTKEY() <> 27
153 RETURN .T.
154 ELSE
155 * Cancel or Escape returns false
156 RETURN .F.
157 ENDIF
158 *: EOF: YESNO.ACT

```

10/27/93	BROWSWIN.SPR	14:49:42
Author's Name Copyright (c) 1993 Company Name Address City, Zip Description: This program was automatically generated by GENSCRN.		

```

1  *
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *

```

```

18 #REGION 0
19 REGIONAL m.curarea, m.talkstat, m.compstat

```

```

20 IF SET("TALK") = "ON"
21 SET TALK OFF
22 m.talkstat = "ON"
23 ELSE
24 m.talkstat = "Off"
25 ENDIF
26 m.compstat = SET("COMPATIBLE")
27 SET COMPATIBLE FOXPLUS

```

MS-DOS Window definitions

```

28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *

```

```

38 IF NOT WEXIST("browsewin") ;
39 OR UPPER(WTITLE("BROWSEWIN")) == "BROWSEWIN.PJX" ;
40 OR UPPER(WTITLE("BROWSEWIN")) == "BROWSEWIN.SCX" ;
41 OR UPPER(WTITLE("BROWSEWIN")) == "BROWSEWIN.MNX" ;
42 OR UPPER(WTITLE("BROWSEWIN")) == "BROWSEWIN.PRG" ;
43 OR UPPER(WTITLE("BROWSEWIN")) == "BROWSEWIN.FRX" ;
44 OR UPPER(WTITLE("BROWSEWIN")) == "BROWSEWIN.QPR" ;
45 DEFINE WINDOW browsewin ;
46 FROM INT((SROW()-21)/2), INT((SCOL()-80)/2) ;
47 TO INT((SROW()-21)/2)+20, INT((SCOL()-80)/2)+79 ;
48 FLOAT ;
49 NOCLOSE ;
50 SHADOW ;
51 minimize ;
52 SYSTEM ;
53 COLOR SCHEME 1
54 ENDIF

```

BROWSWIN/MS-DOS Screen Layout

```

55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *

```

```

64 #REGION 1
65 IF WVISIBLE("browsewin")
66 ACTIVATE WINDOW browsewin SAME

```

```

67 ELSE
68 ACTIVATE WINDOW browsewin NOSHOW
69 ENDIF
70
71 IF NOT WVISIBLE("browsewin")
72 ACTIVATE WINDOW browsewin
73 ENDIF
74 READ CYCLE
75 RELEASE WINDOW browsewin
76
77 #REGION 0
78 IF m.talkstat = "ON"
79 SET TALK ON
80 ENDIF
81 IF m.compstat = "ON"
82 SET COMPATIBLE ON
83 ENDIF
84 *: EOF: BROWSWIN.AC1
85
86

```

10/27/93	HMAT.SPR	14:48:51
Author's Name Copyright (c) 1993 Company Name Address City, Zip Description: This program was automatically generated by GENSCRN.		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *

```

```

18 #REGION 0
19 REGIONAL m.curarea, m.talkstat, m.compstat
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

```

IF SET("TALK") = "ON"
  SET TALK OFF
  m.talkstat = "ON"
ELSE
  m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

MS-DOS window definitions

```

IF NOT VEXIST("hmat") ;
  OR UPPER(WTITLE("HMAT")) = "HMAT.PJX" ;
  OR UPPER(WTITLE("HMAT")) = "HMAT.SCX" ;
  OR UPPER(WTITLE("HMAT")) = "HMAT.MNX" ;
  OR UPPER(WTITLE("HMAT")) = "HMAT.PRG" ;
  OR UPPER(WTITLE("HMAT")) = "HMAT.FRX" ;
  OR UPPER(WTITLE("HMAT")) = "HMAT.QPR" ;
  DEFINE WINDOW hmat ;
  FROM INT((SROW()-21)/2), INT((SCOL()-80)/2) ;
  TO INT((SROW()-21)/2)+20, INT((SCOL()-80)/2)+79 ;
  TITLE "HAZARDOUS MATERIALS" ;
  NOFLOAT ;
  NOCLOSE ;
  SHADOW ;
  nonimize ;
  DOUBLE ;
  COLOR SCHEME 1
ENDIF

```

HMAT/MS-DOS Setup Code - SECTION 2

```

#REGION 1
PUSH KEY

```

```

67 *ON KEY LABEL ESC DO EscPressed
68 m.oldscape = SET( "ESCAPE" )
69 SET ESCAPE OFF
70 m.adding = .f.
71 m.change = .f.
72 *allowedit = .f.
73
74 IF USED('hmat')
75   SELECT hmat
76   SET ORDER TO TAG hmatid
77 ELSE
78   USE hmat AGAIN ORDER TAG hmatid
79 ENDIF
80 *****
81 * Check see if the last record is defined
82 IF TYPE( "m.LastRec" ) = "U"
83
84   * Start with the first record
85   GO TOP
86   m.lastrec = RECNO()
87
88 ELSE
89   * Start on the last record used
90   GO m.lastrec
91 ENDIF
92 *****
93 SCATTER MENVAR MEMO
94 m.hmcom = comame(ALLTRIM(m.niin))
95 m.oldhmcom = m.hmcom
96
97 *
98 *
99 *
100 *
101 *
102 *
103
104 #REGION 1
105 IF WVISIBLE("hmat")
106   ACTIVATE WINDOW hmat SAME
107 ELSE
108   ACTIVATE WINDOW hmat NOSHOW
109 ENDIF
110 @ 0,2 SAY "ID" ;
111   SIZE 1,2, 0 ;
112 @ 2,2 SAY "NIIN #:" ;
113   SIZE 1,7, 0 ;
114 @ 2,23 SAY "Common Name:" ;
115   SIZE 1,12, 0 ;
116 @ 1,2 SAY "Manufa" ;
117   SIZE 1,6, 0 ;
118 @ 0,14 SAY "Materials Name:" ;
119   SIZE 1,15, 0 ;
120 @ 0,5 GET m.hmatid ;
121   SIZE 1,5 ;
122   DEFAULT 0 ;
123   DISABLE
124 @ 0,32 GET m.hmatname ;
125   SIZE 1,44 ;
126   DEFAULT " " ;
127   PICTURE "99" ;
128   VALID _qk4vr594( )
129 @ 1,15 GET m.mfg ;
130   SIZE 1,61 ;
131   DEFAULT " " ;
132   VALID _qk4vr3ek( )

```

HMAT/MS-DOS Screen Layout

```

133 @ 2,10 GET m.niin ;
134 SIZE 1,10 ;
135 DEFAULT " " ;
136 VALID _qkf0vr3hm()
137 @ 2,37 GET m.hmcom ;
138 SIZE 1,39 ;
139 DEFAULT " " ;
140 WHEN _qkf0vr3hq() ;
141 VALID _qkf0vr3me() ;
142 DISABLE
143 @ 4,0 EDIT m.msds ;
144 SIZE 13,78,0 ;
145 DEFAULT " " ;
146 SCROLL ;
147 VALID _qkf0vr3on()
148 @ 18,6 GET m.action ;
149 PICTURE "@*HN \<Add;\<Next;\<Previous;\?E\<xit" ;
150 SIZE 1,10,1 ;
151 DEFAULT 1 ;
152 VALID _qkf0vr3r9()
153 @ 18,51 GET m.save ;
154 PICTURE "@*HN \<Save;\<Cancel" ;
155 SIZE 1,8,1 ;
156 VALID _qkf0vr3xh() ;
157 DISABLE
158 @ 1,8 SAY "cturer:" ;
159 SIZE 1,7,0
160
161 [IF NOT WWISIBLE("hmat")
162 ACTIVATE WINDOW hmat
163 ]
164 ]
165
166 READ CYCLE MODAL
167
168 RELEASE WINDOW hmat
169
170 #REGION 0
171 [IF m.talkstat = "ON"
172 SET TALK ON
173 ]
174 [IF m.compstat = "ON"
175 SET COMPATIBLE ON
176 ]
177 ]
178
179 *
180 *
181 *
182 *
183 *
184 *
185
186 #REGION 1
187 SELECT hmcom
188 USE
189 SELECT hmat
190 USE
191 POP KEY ALL
192 SET ESCAPE &oldescape
193 ***** End of Main Body - Entry Cleanup
194
195 *****
196
197 *
198

```

HMAT/MS-DOS Cleanup Code

```

199 *
200 *
201 *
202 *
203
204 #REGION 1
205 PROCEDURE CHANGE
206 *****
207 m.oldexact = SET( "EXACT" )
208 SET EXACT ON
209 m.change =(TRIM(hmat.hmname) <> TRIM( m.hmname));
210 OR m.mfg <> hmat.mfg;
211 OR m.niin <> hmat.niin;
212 OR m.msds <> hmat.msds)
213 SET EXACT &oldexact
214 RETURN m.change
215
216 *****
217 FUNCTION newline
218 *****
219 PARAMETER TEXT
220 nl = CHR(10) + CHR(13)
221 IF EMPTY(m.text)
222 m.text = ""
223 ]
224 ]
225 ]
226 m.text = m.text + nl + nl
227 ]
228 ]
229 ]
230 FUNCTION commname
231 *****
232 PARAMETER mkey
233 PRIVATE msel
234 msel = SELECT()
235 IF IUSED("HMCOM")
236 SELECT 0
237 ]
238 ]
239 ]
240 ]
241 ]
242 ]
243 ]
244 ]
245 ]
246 ]
247 ]
248 ]
249 ]
250 ]
251 ]
252 ]
253 ]
254 ]
255 ]
256 ]
257 ]
258 ]
259 ]
260 ]
261 ]
262 ]
263 ]
264 ]

```

HMAT/MS-DOS Supporting Procedures and Functions

```

265 IF FOUND()
266   REPLACE niin WITH mkey
267   REPLACE common WITH mname
268 ELSE
269   SET ORDER TO TAG hmcomid
270   GO BOTTOM
271   id = hmcomid + 1
272   REPLACE hmcomid WITH id
273   REPLACE niin WITH mkey
274   REPLACE common WITH mname
275 ENDIF
276 SELECT (msel)
277 RETURN
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330

```

```

      _QKFOVR394      M.hmatname VALID
      Function Origin:
      From Platform:  MS-DOS      Record Number:  8
      From Screen:    HMAT,
      Variable:       M.hmatname
      Called By:      VALID Clause
      Object Type:    Field
      Snippet Number: 1

```

```

      FUNCTION _qkf0vr394      && M.hmatname VALID
      #REGION 1
      IF m.adding = .T.
      m.oldrec = RECNO()
      SET ORDER TO TAG hmatname
      SEEK(m.hmatname)
      IF FOUND()
      =ermsg("Material name already exist",2)
      SCATTER FIELD hmatname MEMVAR BLANK
      CUROBJ = G@JNUM(m.hmatname)
      ENDIF
      SET ORDER TO TAG hmatid
      GO m.oldrec
      ENDIF
      DO CHANGE
      IF m.change
      SHOW GET m.action disabled
      SHOW GET m.save enabled
      ENDIF
      SHOW GETS

```

```

      _QKFOVR394      m.mfg VALID
      Function Origin:
      From Platform:  MS-DOS      Record Number:  9
      From Screen:    HMAT,
      Variable:       m.mfg
      Called By:      VALID Clause
      Object Type:    Field
      Snippet Number: 2

```

```

331 *
332 FUNCTION _qkf0vr3ek      && m.mfg VALID
333 #REGION 1
334 DO CHANGE
335 IF m.change
336   SHOW GET m.action disabled
337   SHOW GET m.save enabled
338 ENDIF
339 SHOW GETS
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396

```

```

      _QKFOVR3HM      m.niin VALID
      Function Origin:
      From Platform:  MS-DOS      Record Number:  10
      From Screen:    HMAT,
      Variable:       m.niin
      Called By:      VALID Clause
      Object Type:    Field
      Snippet Number: 3

```

```

      FUNCTION _qkf0vr3hm      && m.niin VALID
      #REGION 1
      m.hmcom = comname(ALLTRIM(m.niin))
      IF EMPTY(m.hmcom)
      SHOW GET m.hmcom enabled
      ELSE
      SHOW GET m.hmcom disabled
      ENDIF
      DO CHANGE
      IF m.change
      SHOW GET m.action disabled
      SHOW GET m.save enabled
      ENDIF
      SHOW GETS

```

```

      _QKFOVR3KQ      m.hmcom WHEN
      Function Origin:
      From Platform:  MS-DOS      Record Number:  11
      From Screen:    HMAT,
      Variable:       m.hmcom
      Called By:      WHEN Clause
      Object Type:    Field
      Snippet Number: 4

```

```

      FUNCTION _qkf0vr3kq      && m.hmcom WHEN
      #REGION 1
      m.oldhmcom = m.hmcom

```

```

      _QKFOVR3ME      m.hmcom VALID
      Function Origin:
      From Platform:  MS-DOS

```

From Screen: HMAT, Record Number: 11
 Variable: m.hmcom
 Called By: VALID Clause
 Object Type: Field
 Snippet Number: 5

FUNCTION _qkf0vr3me && m.hmcom VALID

#REGION 1
 IF m.hmcom <> m.olchmcom
 SHOW GET m.action disabled
 SHOW GET m.save enabled
 ENDIF

m.msds VALID

Function Origin:

From Platform: MS-DOS
 From Screen: HMAT, Record Number: 12
 Variable: m.msds
 Called By: VALID Clause
 Object Type: Field
 Snippet Number: 6

FUNCTION _qkf0vr3on && m.msds VALID

DO CHANGE

IF m.change
 SHOW GET m.action disabled
 SHOW GET m.save enabled
 ENDIF

m.Action VALID

Function Origin:

From Platform: MS-DOS
 From Screen: HMAT, Record Number: 13
 Variable: m.Action
 Called By: VALID Clause
 Object Type: Push Button
 Snippet Number: 7

FUNCTION _qkf0vr3r9 && m.Action VALID

IF m.action = 1

* SCATTER FIELD hmatname MEMVAR BLANK
 m.hmatid = RECCOUNT()+1
 SHOW GET m.hmatname enabled
 SHOW GET m.mfg enabled
 SHOW GET m.niin enabled
 SHOW GET action disabled
 SHOW GET SAVE enabled
 m.adding = .T.
 SHOW GETS

ELSE

DO CASE
 CASE m.action = 2
 SKIP
 IF EOF()
 ?? CHR(7)
 WAIT "Last record" WINDOW NOWAIT
 GO BOTTOM
 ENDIF

CASE m.action = 3
 SKIP -1
 IF BOF()
 ?? CHR(7)
 WAIT "First record" WINDOW NOWAIT
 GO TOP
 ENDIF

CASE m.action = 4
 CLEAR READ
 ENDCASE

IF m.action > 1 AND m.action < 4
 * allowedit = .f.
 SCATTER MEMVAR MEMO
 m.hmcom = cfname(ALLTRIM(m.niin))
 SHOW GETS
 ENDIF

_QKF0VR3XH m.Save VALID

Function Origin:

From Platform: MS-DOS
 From Screen: HMAT, Record Number: 14
 Variable: m.Save
 Called By: VALID Clause
 Object Type: Push Button
 Snippet Number: 8

* did the record change
 FUNCTION _qkf0vr3xh && m.Save VALID

#REGION 1
 IF m.save = 1 && Selected Save Button
 IF EMPTY(m.hmatname)
 = errmsg("Please enter Materials Name !!!",1)
 CUROBJ = OBJNUM(m.hmatname)
 SHOW GETS
 RETURN

ENDIF

IF EMPTY(m.niin)
 = errmsg("Please enter Stock Number !!!",1)
 CUROBJ = OBJNUM(m.niin)
 SHOW GETS
 RETURN

ENDIF

IF m.adding && Adding a new record

```

529 APPEND BLANK MEMO
530 GATHER MEMVAR MEMO
531 ELSE
532 DO CHANGE
533 IF m.change && Changing an old record
534 GATHER MEMVAR MEMO
535 ENDIF
536 ENDIF
537 DO savecom WITH m.niin, m.hmcom
538 ELSE
539 SCATTER MEMVAR MEMO
540 m.hmcom = coname(ALLTRIM(m.niin))
541 ENDIF
542
543 SHOW GETS
544 SHOW GET m.hmcom disabled
545 SHOW GET action.enabled
546 SHOW GET SAVE disabled
547
548 m.adding = .f.
549 m.change = .f.
550 *: EOF: HMAT.AC1

```


10/27/93	HMCF.SPR	14:49:05
Author's Name Copyright (c) 1993 Company Name Address City, Zip Description: This program was automatically generated by GENSCREEN.		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *

```

```

18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

```

REGION 0
REGIONAL m.curarea, m.talkstat, m.compstat
IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

MS-DOS Window definitions

```

IF NOT HEX(IST("HMCF")) ;
OR UPPER(UTITLE("HMCF")) = "HMCF.PJX" ;
OR UPPER(UTITLE("HMCF")) = "HMCF.SCX" ;
OR UPPER(UTITLE("HMCF")) = "HMCF.MNX" ;
OR UPPER(UTITLE("HMCF")) = "HMCF.PRG" ;
OR UPPER(UTITLE("HMCF")) = "HMCF.FRX" ;
OR UPPER(UTITLE("HMCF")) = "HMCF.QPR" ;
DEFINE WINDOW hmcf ;
FROM INT((SROW()-12)/2), INT((SCOL()-53)/2) ;
TO INT((SROW()-12)/2)+11, INT((SCOL()-53)/2)+52 ;
NOFLOAT ;
NOCLOSE ;
SHADOW ;
NOMINIMIZE ;
DOUBLE ;
COLOR SCHEME 1
ENDIF

```

HMCF/MS-DOS Setup Code - SECTION 2

```

REGION 1
PUSH KEY
"ON KEY LABEL ESC DO EscPressed

```

```

67 m.oldscape = SET( "ESCAPE" )
68 SET ESCAPE OFF
69 m.adding = .F.
70 m.change = .F.
71
72 CLOSE ALL
73 SELECT 0
74 USE hmcf
75 SET ORDER TO TAG hmcf OF hmcf.cdx
76
77 *****
78 * Check see if the last record is defined
79 IF TYPE( "m.LastRec" ) = "U"
80
81 * Start with the first record
82 GO TOP
83 m.lastrec = RECNO()
84
85 ELSE
86 * Start on the last record used
87 GO m.lastrec
88
89 *****
90 SCATTER MEMVAR
91
92
93
94
95
96
97
98
99
100 REGION 1
101 IF LVISIBLE("hmcf")
102   ACTIVATE WINDOW hmcf SAME
103 ELSE
104   ACTIVATE WINDOW hmcf NOSHOW
105 ENDIF
106 @ 1,40 GET m.action ;
107 PICTURE "99VN \<Add;\<Edit;\<Previous;\<Next;\<Previous;\<Next" ;
108 SIZE 1,10,1 ;
109 DEFAULT 1 ;
110 VALID qkfbvrdyq()
111 @ 2,14 GET m.hmcfid ;
112 SIZE 1,10 ;
113 DEFAULT 0 ;
114 DISABLE
115 @ 4,14 GET m.hmcf ;
116 SIZE 1,25 ;
117 DEFAULT " " ;
118 PICTURE "9i" ;
119 VALID qkfbvre5a() ;
120 DISABLE
121 @ 8,11 GET m.save ;
122 PICTURE "99HN \<Save;\<Cancel" ;
123 SIZE 1,8,1 ;
124 DEFAULT 1 ;
125 VALID qkfbvre7o() ;
126 DISABLE
127 @ 1,0 TO 9,39
128 @ 0,11 SAY "HM COST FACTORS" ;
129 SIZE 1,15,0
130 @ 2,5 SAY "ID NUM:" ;
131 SIZE 1,7,0
132 @ 4,1 SAY "COST FACTOR:" ;

```

HMCF/MS-DOS Screen Layout

SIZE 1,12, 0

```

133 IF NOT UNVISIBLE("hmcf")
134   ACTIVATE WINDOW hmcf
135   ENDIF
136
137
138 READ CYCLE
139
140 RELEASE WINDOW hmcf
141
142 #REGION 0
143 IF m.talkstat = "ON"
144   SET TALK ON
145   ENDIF
146 IF m.compatstat = "ON"
147   SET COMPATIBLE ON
148   ENDIF
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173

```

HMCF/MS-DOS Cleanup Code

```

174 #REGION 1
175 POP KEY ALL
176 SET ESCAPE &ldescape
177 ***** End of Main Body - Entry Cleanup
178 *****
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198

```

HMCF/MS-DOS Supporting Procedures and Functions

```

199 #REGION 1
200 PROCEDURE CHANGE
201 *****
202 m.oldexact = SET( "EXACT" )
203 SET EXACT ON
204 m.change = (TRIM(hmcf.hmcf) <> TRIM( m.hmcf))
205 SET EXACT &ldexact
206 RETURN m.change
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264

```

_QKFOVRDYO m.Action VALID

Function Origin:

From Platform: MS-DOS
 From Screen: HMCF,
 Variable: m.Action
 Called By: VALID Clause
 Object Type: Push Button
 Snippet Number: 1

Record Number: 4

```

199 FUNCTION _qkfovrdoq && m.Action VALID
200 #REGION 1
201 IF m.action = 1
202   SCATTER MEMVAR BLANK
203   m.hmcfid=RECCOUNT()+1
204   SHOW GETS
205   SHOW GET m.hmcf enabled
206   SHOW GET action disabled
207   SHOW GET SAVE enabled
208   m.adding = .T.
209 ELSE
210   --OO CASE
211   --CASE m.action = 2
212   SHOW GETS
213   SHOW GET action disabled
214   SHOW GET SAVE enabled
215   SHOW GET m.hmcf enabled
216   --CASE m.action = 3
217   SKIP
218   IF EOF()
219     77 CHR( 7 )
220     WAIT "Last record" WINDOW NOWAIT
221     SKIP -1
222   ELSE
223     SCATTER MEMVAR
224     SHOW GETS
225   ENDIF
226   --CASE m.action = 4
227   SKIP -1
228   IF BOF()
229     77 CHR( 7 )
230     WAIT "First record" WINDOW NOWAIT
231     SKIP
232   ELSE
233     SCATTER MEMVAR
234     SHOW GETS
235   ENDIF
236   --CASE m.action = 5
237   CLEAR READ
238   ENDIF
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264

```

_QKFOVRESA m.hmcf VALID

Function Origin:

From Platform: MS-DOS
 From Screen: HMCF,
 Variable: m.hmcf
 Called By: VALID Clause
 Object Type: Field
 Snippet Number: 2

Record Number: 6

```

265 FUNCTION _qkfovrdoq && m.hmcf VALID
266 #REGION 1

```

```

265 IF m.adding
266   m.oldrec = RECMO()
267   SEEK m.hmcf
268   IF FOUND()
269     =errmsg("Record already exists",1)
270     SCATTER MEMVAR BLANK FIELD hmcf
271   ENDIF
272   GO m.oldrec
273 ENDIF
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291 FUNCTION _qkf0vre7o  && m.Save VALID
292 #REGION 1
293   m.notsave = .F.
294   IF m.save = 1 && Selected Save Button
295     =errmsg("Data empty, could not save!",2)
296     CURSOR = OBJNUM(m.hmcf)
297     m.notsave = .T.
298   ELSE
299     IF m.adding && Adding a new record
300       APPEND BLANK
301       GATHER MEMVAR
302     ELSE
303       IF m.change && Changing an old record
304         GATHER MEMVAR
305       ENDIF
306     ENDIF
307   ELSE
308     SCATTER MEMVAR
309   ENDIF
310
311 SHOW GETS
312 IF m.notsave
313   SHOW GET m.hmcf enabled
314   SHOW GET action disabled
315   SHOW GET SAVE enabled
316 ELSE
317   SHOW GET m.hmcf disabled
318   SHOW GET action enabled
319   SHOW GET SAVE disabled
320   m.adding = .F.
321   m.change = .F.
322 ENDIF
323
324
325
326

```

_QKFOVRE7O	m.Save VALID
Function Origin:	
From Platform:	MS-DOS
From Screen:	HMCF,
Variable:	m.Save
Called By:	VALID Clause
Object Type:	Push Button
Snippet Number:	3
	Record Number: 7

10/27/93	HMCFE.SPR	14:49:08
Author's Name Copyright (c) 1993 Company Name Address City, Zip Description: This program was automatically generated by GENSCRN.		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

```

19 #REGION 0
20 REGIONAL m.curarea, m.talkstat, m.compstat
21
22 IF SET("TALK") = "ON"
23   SET TALK OFF
24   m.talkstat = "ON"
25 ELSE
26   m.talkstat = "OFF"
27 ENDIF
28 m.compstat = SET("COMPATIBLE")
29 SET COMPATIBLE FOXPLUS
30
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

MS-DOS Window definitions

```

38 IF NOT WEXIST("hmcfe");
39   OR UPPER(UTITLE("HMCFE")) = "HMCFE.PJX";
40   OR UPPER(UTITLE("HMCFE")) = "HMCFE.SCX";
41   OR UPPER(UTITLE("HMCFE")) = "HMCFE.MHX";
42   OR UPPER(UTITLE("HMCFE")) = "HMCFE.PRG";
43   OR UPPER(UTITLE("HMCFE")) = "HMCFE.FRX";
44   OR UPPER(UTITLE("HMCFE")) = "HMCFE.QPR"
45   DEFINE WINDOW hmcfe;
46   FROM INT((SROW()-16)/2), INT((SCOL()-67)/2) ;
47   TO INT((SROW()-16)/2)+15, INT((SCOL()-67)/2)+66 ;
48   NOFLOAT ;
49   NOCLOSE ;
50   SHADOW ;
51   NOMINIMIZE ;
52   DOUBLE ;
53   COLOR SCHEME 1
54 ENDIF
55
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

HMCFE/MS-DOS Setup Code - SECTION 2

```

64 #REGION 1
65 PUSH .EY
66 *ON KEY LABEL ESC DO EscPressed

```

```

67 m.oldscape = SET( "ESCAPE" )
68 SET ESCAPE OFF
69 m.adding = .f.
70 m.change = .f.
71 m.action=5
72 CLOSE ALL
73
74 IF USED("HMCFE")
75   SELECT hmcfe
76   SET ORDER TO hmcfid
77 ELSE
78   SELECT 0
79   USE (LOCFILE("HMCFE.dbf","DBF","where is hmcfe?"));
80   AGAIN ALIAS hmcfe ;
81   ORDER hmcfid
82 ENDIF
83
84 *USE HMCFE AGAIN
85 *****
86 * Check see if the last record is defined
87 GO TOP
88 *****
89 SCATTER MENVAR
90 IF m.hmcfid>0
91   m.answr=get_hmcf(m.hmcfid)
92 ELSE
93   m.answr=""
94 ENDIF
95
96 m.oldscur=m.answr
97 SHOW GETS
98
99 *
100 *
101 *
102 *
103 *
104 *
105 *
106 *
107 *
108 *
109 *
110 #REGION 1
111 IF UNVISIBLE("hmcfe")
112   ACTIVATE WINDOW hmcfe SAME
113 ELSE
114   ACTIVATE WINDOW hmcfe NOSHOW
115 ENDIF
116 @ 1,54 GET m.action ;
117 PICTURE "a*VN \<Add;\<Next;\<Previous;\<Top;\<Bottom;\<E
=>\<xit" ;
118
119 SIZE 1,10,1 ;
120 DEFAULT 1 ;
121 VALID _qkf0vrggz()
122 @ 5,3 GET m.factor ;
123 PICTURE "a*1Hk " ;
124 SIZE 1,10,1 ;
125 DEFAULT 0 ;
126 WHEN _qkf0vrggz() ;
127 VALID _qkf0vrgtp()
128 @ 7,3 GET m.element ;
129 PICTURE "a*1Hk " ;
130 SIZE 1,10,1 ;
131 DEFAULT 0 ;
132 WHEN _qkf0vrggz4() ;
133 VALID _qkf0vrgvq()
134
135 *
136 *
137 *
138 *
139 *
140 *
141 *
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 *
152 *
153 *
154 *
155 *
156 *
157 *
158 *
159 *
160 *
161 *
162 *
163 *
164 *
165 *
166 *

```

HMCFE/MS-DOS Screen Layout


```

264 *****
265 PARAMETERS m.answr
266 IF PARAMETER()=0
267 X=0
268 ELSE
269 m.oldfile=SELECT()
270
271 SELECT DISTINCT hmcfeid;
272 FROM hmcfe;
273 WHERE hmcfeid IN (m.answr);
274 INTO ARRAY X
275 X=X||
276 SELECT (m.oldfile)
277 ENDIF
278 RETURN X
279
280 *****
281 PROCEDURE rel
282 * DISPLAY RELATIONSHIP
283 * OF HMCFE AND HMCF
284 *****
285 PARAMETER m.id
286 SET TALK OFF
287
288 IF PARAMETER()=0
289 m.id=0
290 ENDIF
291
292 m.oldfile=SELECT()
293
294 &&OPEN FILE # 1
295
296 IF USED("HMCFEI")
297 SELECT hmcfeid
298 SET ORDER TO hmcfeid
299
300 ELSE
301 SELECT 0
302 USE hmcfe ORDER hmcfeid
303 ENDIF
304
305 &&OPEN FILE # 2
306
307 IF USED("HMCFE")
308 SELECT hmcfe
309 SET ORDER TO hmcfeid
310
311 ELSE
312 SELECT 0
313 USE hmcfe AGAIN ORDER hmcfeid
314 ENDIF
315
316 &&OPEN FILE #3
317
318 IF USED("HMCFF")
319 SELECT hmcff
320 SET ORDER TO hmcffid
321
322 ELSE
323 SELECT 0
324 USE hmcff ORDER hmcffid
325 ENDIF
326
327 IF m.id>0
328 SET FILTER TO hmcfeid = m.id
329 ENDIF

```

```

330 SELECT hmcfe
331 SET RELATION TO hmcfeid INTO hmcfe ADDITIVE
332
333 SELECT hmcfe
334
335 SET RELATION TO hmcfeid INTO hmcfe ADDITIVE
336
337 SET SKIP TO hmcfeid, hmcfe
338
339 ** Show fields from grandparent (HMCFEI), parent (HMCFF) and child (H
=> MCF)**
340 BROWSE FIELDS hmcfeid;H="FACTORS", hmcfeid;H="ELEMENTS",;
341 hmcfeid;H="ITEMS", hmcfeid;H="COSTS";
342 FOR hmcfeid = hmcfeid;HMCFFID NOAPPEND NODELETE NORMAL
=> TITLE "COST FACTORS"
343
344 m.x=hmcfeid.hmcfeid
345 SET FILTER TO
346 SELECT hmcfe
347 SET RELATION TO
348 SELECT hmcfe
349 SET RELATION TO
350 SELECT (m.oldfile)
351 RETURN m.x
352
353 *
354 *
355 *
356 *
357 *
358 *
359 *
360 *
361 *
362 *
363 *
364 *
365 *
366 *
367 *
368 *
369 *
370 *
371 *
372 *
373 *
374 *
375 *
376 *
377 *
378 *
379 *
380 *
381 *
382 *
383 *
384 *
385 *
386 *
387 *
388 *
389 *
390 *
391 *
392 *
393 *

```

_QKF0VRGGZ		m.Action VALID	
Function Origin:			
From Platform:	MS-DDS	Record Number:	4
From Screen:	HMCFF	m.Action	
Variable:	VALID Clause	Push Button	
Called By:			
Object Type:			
Snippet Number:	1		

```

FUNCTION qkf0vrvgz && m.Action VALID
#REGION 1
IF m.action = 1
GO BOTTOM
SCATTER MEMVAR FIELD hmcfe,hmcfeid BLANK
m.hmcfeid = RECCOUNT()+1
SHOW GETS
SHOW GET action disabled
SHOW GET SAVE enabled
SHOW GET m.answr enabled
SHOW GET m.hmcfeid enabled
SHOW GET m.factor disabled
SHOW GET m.element disabled
m.adding = .T.
ELSE
--DO CASE
--CASE m.action = 2
SHOW GETS
SHOW GET action disabled
SHOW GET SAVE enabled
SHOW GET m.answr enabled
SHOW GET m.hmcfeid enabled
SHOW GET m.factor disabled

```

```

394 SHOW GET m.element disabled
395
396 CASE m.action = 3
397 SKIP
398 IF EOF()
399 ?? CHR( 7 )
400 WAIT "Last record" WINDOW NOWAIT
401 SKIP -1
402 ENDF
403
404 CASE m.action = 4
405 SKIP -1
406 IF BOF()
407 ?? CHR( 7 )
408 WAIT "First record" WINDOW NOWAIT
409 SKIP
410 ENDF
411
412 CASE m.action = 5
413 GO TOP
414
415 CASE m.action = 6
416 GO BOTTOM
417
418 CASE m.action = 7
419 CLEAR READ
420 ENDCASE
421
422 IF m.action > 2 AND m.action < 7
423 SCATTER MENVAR
424 IF m.hmcfid>0
425 m.answr=get_hmc(m.hmcfid)
426 m.oldanswr=m.answr
427 ELSE
428 m.answr=""
429 m.oldanswr=""
430 ENDF
431 SHOW GETS
432 SHOW GET m.hmcfe disabled
433 SHOW GET m.answr disabled
434 SHOW GET m.factor enabled
435 SHOW GET m.element enabled
436 ENDF
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459

```

```

_QKFOVRGRP      m.factor WHEN
Function Origin:
From Platform:  MS-DOS
From Screen:    HMCFE,
Variable:       m.factor
Called By:      WHEN Clause
Snippet Number: 2

```

```

FUNCTION _qkf0vrgrp    && m.factor WHEN
#REGION 1
m.factor=0

```

```

460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525

```

```

_QKFOVRGTP      m.factor VALID
Function Origin:
From Platform:  MS-DOS
From Screen:    HMCFE,
Variable:       m.factor
Called By:      VALID Clause
Snippet Number: 3

```

```

FUNCTION _qkf0vrgrp    && m.factor VALID
#REGION 1
m.factor=rel(0)
IF m.factor>0
LOCATE FOR hmcfeid = m.factor
IF FOUND()
GO RECHO()
SCATTER MENVAR
IF m.hmcfid>0
m.answr=get_hmc(m.hmcfid)
ELSE
m.answr=""
ENDIF
m.oldanswr=m.answr
SHOW GETS
ENDIF

```

```

_QKFOVRGX4      m.element WHEN
Function Origin:
From Platform:  MS-DOS
From Screen:    HMCFE,
Variable:       m.element
Called By:      WHEN Clause
Snippet Number: 4

```

```

FUNCTION _qkf0vrngx4    && m.element WHEN
#REGION 1
m.element=m.hmcfid

```

```

_QKFOVRGYG      m.element VALID
Function Origin:
From Platform:  MS-DOS
From Screen:    HMCFE,
Variable:       m.element
Called By:      VALID Clause
Snippet Number: 5

```

```

FUNCTION _qkf0vrgyg    && m.element VALID

```

```

526 #REGION 1
527 m.element=rel(m.hmcfid)
528 IF m.element>0
529 LOCATE FOR hmcfeid = m.element
530
531 IF FOUND()
532 GO RECNO()
533 SCATTER MEMVAR
534 IF m.hmcfid>0
535 m.answr=get_hmcf(m.hmcfid)
536 ELSE
537 m.answr=""
538 ENDIF
539 m.oldanswr=m.answr
540 SHOW GETS
541 ENDIF
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591

```

```

_QKFOVRH1G
Function Origin:
From Platform: MS-DOS
From Screen: HMCFE,
Variable: m.answr
Called By: WHEN Clause
Object Type: Field
Snippet Number: 6
Record Number: 8

```

```

_QKFOVRH34
Function Origin:
From Platform: MS-DOS
From Screen: HMCFE,
Variable: m.answr
Called By: VALID Clause
Object Type: Field
Snippet Number: 7
Record Number: 8

```

```

592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641

```

```

642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

_QKFOVRH6D
Function Origin:
From Platform: MS-DOS
From Screen: HMCFE,
Variable: m.answr
Called By: VALID Clause
Object Type: Push Button
Snippet Number: 8
Record Number: 11

```

```

999

```


10/27/93	HMCFEI.SPR	14:49:13
Author's Name Copyright (c) 1993 Company Name Address City, Zip Description: This program was automatically generated by GENSCRN.		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *

```

```

18 #REGION 0
19 REGIONAL m.curarea, m.talkstat, m.compstat
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

```

```

39 IF SET("TALK") = "ON"
40 SET TALK OFF
41 m.talkstat = "ON"
42 ELSE
43 m.talkstat = "OFF"
44 ENDIF
45 m.compstat = SET("COMPATIBLE")
46 SET COMPATIBLE FOXPLUS
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

MS-DOS Window definitions

```

67 IF NOT EXIST("hmcfei") ;
68 OR UPPER(WTITLE("HMCFEI")) = "HMCFEI.PJX" ;
69 OR UPPER(WTITLE("HMCFEI")) = "HMCFEI.SCX" ;
70 OR UPPER(WTITLE("HMCFEI")) = "HMCFEI.MNX" ;
71 OR UPPER(WTITLE("HMCFEI")) = "HMCFEI.PRG" ;
72 OR UPPER(WTITLE("HMCFEI")) = "HMCFEI.FRX" ;
73 OR UPPER(WTITLE("HMCFEI")) = "HMCFEI.QPR" ;
74 DEFINE WINDOW hmcfei ;
75 FROM INT((SROW()-17)/2), INT((SCOL()-69)/2) ;
76 TO INT((SROW()-17)/2)+16, INT((SCOL()-69)/2)+68 ;
77 NOFLOAT ;
78 NOCLOSE ;
79 SHADOW ;
80 NOMINIMIZE ;
81 DOUBLE ;
82 COLOR SCHEME 1
83 ENDIF
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131

```

HMCFEI/MS-DOS Setup Code - SECTION 2

```

132 #REGION 1
133 PUSH KEY
134 *ON KEY LABEL ESC DO EscPressed
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166

```

```

67 m.oldscape = SET("ESCAPE")
68 SET ESCAPE OFF
69 m.adding = .F.
70 m.change = .F.
71 m.editing = .F.
72 m.action=5
73 CLOSE ALL
74
75 IF USED("HMCFEI")
76 SELECT hmcfei
77 SET ORDER TO hmcfid
78 ELSE
79 SELECT 0
80 USE (LOCFILE("HMCFEI.dbf","DBF","Where is HMCFEI?"));
81 AGAIN ALIAS hmcfei ;
82 ORDER hmcfid
83 ENDIF
84
85 *USE HMCFF AGAIN
86 *****
87 * Check see if the last record is defined
88 GO TOP
89 *****
90 SCATTER MEMVAR
91 IF m.hmcfid>0
92 m.ansur=get_hmcfe(m.hmcfid)
93 ELSE
94 m.ansur=""
95 ENDIF
96 IF m.hmcfeid>0
97 m.hmcfe=get_hmcfe(m.hmcfeid)
98 ELSE
99 m.hmcfe=""
100 ENDIF
101 m.oldscape=m.hmcfe
102 m.oldsur=m.ansur
103 m.action = 1
104 SHOW GETS
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131

```

HMCFEI/MS-DOS Screen Layout

```

132 #REGION 1
133 IF WISABLE("hmcfei")
134 ACTIVATE WINDOW hmcfei SAME
135 ELSE
136 ACTIVATE WINDOW hmcfei NOSHOW
137 ENDIF
138 @ 0,55 GET m.action ;
139 PICTURE "g*VN \<Add:\<Edit;\<Next;\<Previous;\<Top;\<Bottom;B\<
140 => rowse;\?E\<xit" ;
141 SIZE 1,10,1 ;
142 DEFAULT 1 ;
143 VALID _qkf0vrk34 ;
144 @ 3,2 GET m.factor ;
145 PICTURE "g*1NN " ;
146 SIZE 1,10,1 ;
147 DEFAULT 0 ;
148 WHEN _qkf0vrkgo() ;
149 VALID _qkf0vrk1q()
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166

```

```

132 a 5,2 GET m.element ;
133 PICTURE "g1HN " ;
134 SIZE 1,10,1 ;
135 DEFAULT 0 ;
136 WHEN _qkfvrkmh() ;
137 VALID _qkfvrknv() ;
138 DISABLE
139 a 3,13 GET m.hmcfid ;
140 SIZE 1,5 ;
141 DEFAULT " " ;
142 DISABLE
143 a 3,20 GET m.answr ;
144 SIZE 1,30 ;
145 DEFAULT " " ;
146 WHEN _qkfvrks5() ;
147 VALID _qkfvrktm() ;
148 DISABLE
149 a 5,13 GET m.hmcfeid ;
150 SIZE 1,5 ;
151 DEFAULT 0 ;
152 DISABLE
153 a 5,20 GET m.hmcfe ;
154 SIZE 1,30 ;
155 DEFAULT " " ;
156 PICTURE "g1" ;
157 WHEN _qkfvrkwj() ;
158 VALID _qkfvrkxz() ;
159 DISABLE
160 a 7,13 GET m.hmcfeid ;
161 SIZE 1,5 ;
162 DEFAULT 0 ;
163 DISABLE
164 a 7,20 GET m.hmcfeid ;
165 SIZE 1,30 ;
166 DEFAULT " " ;
167 PICTURE "g1" ;
168 DISABLE
169 a 9,14 GET m.hmcfeicost ;
170 SIZE 1,10 ;
171 DEFAULT 0 ;
172 PICTURE "g3" ;
173 DISABLE
174 a 12,19 GET m.save ;
175 PICTURE "g1HN \<Save;\<Cancel" ;
176 SIZE 1,8,1 ;
177 DEFAULT 1 ;
178 VALID _qkfvr11j() ;
179 DISABLE
180 a 1,0 TO 14,53
181 a 0,14 SAY "COST FACTOR ELEMENT ITEMS" ;
182 SIZE 1,25,0
183 a 5,4 SAY "ELEMENT:" ;
184 SIZE 1,8,0
185 a 3,5 SAY "FACTOR:" ;
186 SIZE 1,7,0
187 a 7,7 SAY "ITEM:" ;
188 SIZE 1,5,0
189 a 9,7 SAY "COST:" ;
190 SIZE 1,5,0
191
192 IF NOT WISIBLE("hmcfeid")
193 [
194 ACTIVATE WINDOW hmcfeid
195 ENDIF
196
197 READ CYCLE

```

```

198 RELEASE WINDOW hmcfeid
199
200 #REGION 0
201 IF m.talkstat = "ON"
202 SET TALK ON
203 ENDIF
204 IF m.compstat = "ON"
205 SET COMPATIBLE ON
206 ENDIF
207
208 *
209 *
210 *
211 *
212 *
213 *
214 *
215
216 #REGION 1
217 POP KEY ALL
218 SET ESCAPE &oldescape
219 ***** End of Main Body - Entry Cleanup *****
220 *****
221
222 *
223 *
224 *
225 *
226 *
227 *
228 *
229
230 #REGION 1
231 PROCEDURE escaped
232 *****
233 RETURN
234
235 *****
236 PROCEDURE CHANGE
237 *****
238 m.oldexact = SET( "EXACT" )
239 SET EXACT ON
240 m.change = (TRIM(hmcfeid.hmcfeid) <> TRIM( m.hmcfeid ))
241 OR hmcfeid.hmcfeid <> m.hmcfeid;
242 OR hmcfeid.hmcfeino <> m.hmcfeino;
243 OR hmcfeid.hmcfeicost <> m.hmcfeicost)
244 SET EXACT &oldexact
245 RETURN m.change
246
247 *****
248 PROCEDURE get hmcfe
249 *****
250 PARAMETER m.hmcfeid
251 m.oldfile=SELECT()
252 DIMENSION X[1]
253
254 SELECT DISTINCT hmcfe.hmcfe;
255 FROM hmcfe;
256 WHERE hmcfe.hmcfeid = (m.hmcfeid);
257 INTO ARRAY X
258 IF NOT EMPTY(X[1])
259 m.hmcfe=X[1]
260 ELSE
261 m.hmcfe = ""
262
263

```

HMCFEI/MS-DOS Cleanup Code

HMCFEI/MS-DOS Supporting Procedures and Functions

```

264 _ENDIF
265
266 SELECT(m.oldfile)
267 RETURN m.hmcfe
268 *****
269 PROCEDURE get_cf
270 *****
271 PARAMETER m.answr
272 DIMENSION hcf[1]
273 hcf[1]=" "
274 m.oldfile=SELECT()
275
276 SELECT DISTINCT hmcfe.hmcfe;
277 FROM hmcfe;
278 WHERE hmcfe.hmcfe IN (m.answr);
279 ORDER BY hmcfe.hmcfe;
280 INTO ARRAY hcf
281
282 _IF NOT EMPTY(hcf[1])
283
284 m.answr=chooser(hcf,"Select a Cost Factor")
285
286 _ELSE
287 =errmsg(m.answr + " was not found",1)
288 m.hmcfeid=0
289 m.answr=" "
290
291 _ENDIF
292
293 SELECT(m.oldfile)
294 RETURN m.answr
295
296 *****
297 PROCEDURE get_cfid
298 *****
299 PARAMETERS m.answr
300 _IF PARAMETER( )=0
301 X=0
302 _ELSE
303 DIMENSION X[1]
304 m.oldfile=SELECT()
305
306 SELECT DISTINCT hmcfe.hmcfeid;
307 FROM hmcfe;
308 WHERE hmcfe.hmcfe IN (m.answr);
309 INTO ARRAY X
310
311 _IF NOT EMPTY(X[1])
312 X=X[1]
313 _ELSE
314 X = 0
315 _ENDIF
316 SELECT (m.oldfile)
317 RETURN X
318
319 *****
320 * COST FACTOR ELEMENTS -- CFE
321 *****
322
323 *****
324 PROCEDURE get_hmcfe
325 *****
326 PARAMETER m.hmcfeid
327 DIMENSION X[1]
328 m.oldfile=SELECT()
329

```

```

330 SELECT DISTINCT hmcfe.hmcfe;
331 FROM hmcfe;
332 WHERE hmcfe.hmcfeid = (m.hmcfeid);
333 INTO ARRAY X
334
335 _IF EMPTY(X[1])
336 m.hmcfe=X[1]
337 _ELSE
338 m.hmcfe = " "
339 _ENDIF
340
341 SELECT(m.oldfile)
342 RETURN m.hmcfe
343 *****
344 PROCEDURE get_cfe
345 *****
346 PARAMETER m.hmcfe
347 DIMENSION hcf[1]
348 hcf[1]=" "
349 m.oldfile=SELECT()
350
351 SELECT DISTINCT hmcfe.hmcfe;
352 FROM hmcfe;
353 WHERE hmcfe.hmcfe IN (m.hmcfe);
354 ORDER BY hmcfe.hmcfeid;
355 INTO ARRAY hcf
356
357 _IF NOT EMPTY(hcf[1])
358
359 m.hmcfe=chooser(hcf,"Select a Cost Factor")
360
361 _ELSE
362 =errmsg(m.hmcfe + " was not found",1)
363 m.hmcfeid=0
364 m.hmcfe=" "
365 _ENDIF
366
367 SELECT(m.oldfile)
368 RETURN m.hmcfe
369
370 *****
371 PROCEDURE get_cfeid
372 *****
373 PARAMETERS m.hmcfe
374 _IF PARAMETER( )=0
375 X=0
376 _ELSE
377 DIMENSION X[1]
378 m.oldfile=SELECT()
379
380 SELECT DISTINCT hmcfe.hmcfeid;
381 FROM hmcfe;
382 WHERE hmcfe.hmcfe IN (m.hmcfe);
383 INTO ARRAY X
384
385 _IF EMPTY(X[1])
386 X=X[1]
387 _ELSE
388 X = 0
389 _ENDIF
390 SELECT (m.oldfile)
391 RETURN X
392
393 *****
394 PROCEDURE rel
395 * DISPLAY RELATIONSHIP

```

```

396 * OF HMCFE AND HMCFC
397 *****
398 PARAMETER m.id
399
400 IF PARAMETER()=0
401   m.id=0
402 ENDIF
403
404 m.oldfile=SELECT()
405 SET TALK OFF
406
407 &&OPEN FILE # 1
408
409 IF USED("HMCFC")
410   SELECT hmcfc
411   SET ORDER TO TAG hmcfcid OF hmcfc.cdx
412 ELSE
413   SELECT 0
414   USE hmcfc ALIAS hmcfc AGAIN
415   SET ORDER TO TAG hmcfcid OF hmcfc.cdx
416 ENDIF
417
418 &&OPEN FILE # 2
419
420 IF USED("HMCFC")
421   SELECT hmcfc
422   SET ORDER TO TAG hmcfcid OF hmcfc.cdx
423 ELSE
424   SELECT 0
425   USE hmcfc ALIAS hmcfc AGAIN
426   SET ORDER TO TAG hmcfcid OF hmcfc.cdx
427 ENDIF
428
429 &&OPEN FILE #3
430
431 IF USED("HMCFC")
432   SELECT hmcfc
433   SET ORDER TO TAG hmcfcid OF hmcfc.cdx
434 ELSE
435   SELECT 0
436   USE hmcfc ALIAS hmcfc AGAIN
437   SET ORDER TO TAG hmcfcid OF hmcfc.cdx
438 ENDIF
439
440 IF m.id>0
441   SET FILTER TO hmcfcid = m.id
442 ENDIF
443
444 SELECT hmcfc
445 SET RELATION TO hmcfcid INTO hmcfc ADDITIVE
446
447 SELECT hmcfc
448 SET RELATION TO hmcfcid INTO hmcfc ADDITIVE
449
450 SET SKIP TO hmcfcid, hmcfc
451
452
453
454
455
456
457
458
459
460
461

```

```

462 ** Show fields from grandparent (HMCFC), parent (HMCFC) and child (M
463 => MCF)**
464 BROWSE FIELDS hmcfc.hmcfc:H="FACTORS", hmcfc.hmcfc:H="ELEMENTS",;
465 hmcfc.hmcfc:H="ITEMS", hmcfc.hmcfc:H="COSTS";
466 FOR hmcfc.hmcfcid = hmcfc.hmcfcid NOMODIFY NOAPPEND NODELETE TITLE "
467 => COST FACTORS";
468 NORMAL
469
470 m.x=hmcfc.hmcfcid
471
472 SET FILTER TO
473 SELECT hmcfc
474 SET RELATION TO
475 SELECT hmcfc
476 SET RELATION TO
477 SELECT(m.oldfile)
478 RETURN m.x
479 *****
480 PROCEDURE browseitem
481 * DISPLAY RELATIONSHIP
482 * OF HMCFC AND HMCFC
483 *****
484 PARAMETER m.id
485
486 IF PARAMETER()=0
487   m.id=0
488 ENDIF
489
490 m.oldfile=SELECT()
491 SET TALK OFF
492
493 &&OPEN FILE # 1
494
495 IF USED("HMCFC")
496   SELECT hmcfc
497   SET ORDER TO TAG hmcfcid OF hmcfc.cdx
498 ELSE
499   SELECT 0
500   USE hmcfc ALIAS hmcfc AGAIN
501   SET ORDER TO TAG hmcfcid OF hmcfc.cdx
502 ENDIF
503
504 &&OPEN FILE # 2
505
506 IF USED("HMCFC")
507   SELECT hmcfc
508   SET ORDER TO TAG hmcfcid OF hmcfc.cdx
509 ELSE
510   SELECT 0
511   USE hmcfc ALIAS hmcfc AGAIN
512   SET ORDER TO TAG hmcfcid OF hmcfc.cdx
513 ENDIF
514
515 &&OPEN FILE #3
516
517 IF USED("HMCFC")
518   SELECT hmcfc
519   SET ORDER TO TAG hmcfcid OF hmcfc.cdx
520 ELSE
521   SELECT 0
522   USE hmcfc ALIAS hmcfc AGAIN
523   SET ORDER TO TAG hmcfcid OF hmcfc.cdx
524 ENDIF
525

```

```
526 ELSE
527     SELECT 0
528     USE hmcfe ALIAS hmcfe AGAIN
529     SET ORDER TO hmcfeid OF hmcfe.cdx
530     ENDIF
531
532 IF m.id>0
533     SET FILTER TO hmcfeid = m.id
534     ENDIF
535
536 SELECT hmcfe
537 SET RELATION TO hmcfeid INTO hmcfeid && ADDITIVE
538
539 SELECT hmcfe
540 SET RELATION TO hmcfeid INTO hmcfe ADDITIVE
541 SET SKIP TO hmcfeid, hmcfe
542
543 ** Show fields from grandparent (HMCFEI), parent (HMCFE) and child (H
544 => HCF)**
545 BROWSE FIELDS hmcfe:h="FACTORS", hmcfe:hmcfeid="ELEMENTS", ;
546 hmcfeid:hmcfeid="ITEMS", hmcfeid:hmcfeid="COSTS",
547 FOR EMPTY(hmcfeid.hmcfeid) NOMODIFY NOAPPEND NODELETE TITLE "COS
548 => T FACTORS".
549 NORMAL
550 m.x=hmcfeid.hmcfeid
551
552 SET FILTER TO
553 SET RELATION TO
554 SELECT(m.oldfile)
555 RETURN m.x
556
557 *
558 *
559 *
560 *
561 *
562 *
563 *
564 *
565 *
566 *
567 *
568 *
569 *
570 *
571 *
572 *
573 *
574 *
575 *
576 *
577 *
578 *
579 *
580 *
581 *
582 *
583 *
584 *
585 *
586 *
587 *
588 *
589 *
```

OKFOVRK34	m.Action VALID
Function Origin:	MS-DOS
From Platform:	HMCFEI,
From Screen:	m.Action
Variable:	VALID Clause
Called By:	Push Button
Object Type:	1
Shippet Number:	2

```
590 SHOW GET m.hmcfeid enabled
591 SHOW GET m.hmcfeidno enabled
592 SHOW GET m.hmcfeicost enabled
593 SHOW GET m.element enabled
594 m.adding = .T.
595 ELSE
596 DO CASE
597 CASE m.action = 2
598     SHOW GETS
599     SHOW GET action disabled
600     SHOW GET m.element enabled
601     SHOW GET SAVE enabled
602     SHOW GET m.answr enabled
603     SHOW GET m.hmcfe enabled
604     SHOW GET m.hmcfeidno enabled
605     SHOW GET m.hmcfeicost enabled
606     m.editing = .T.
607 CASE m.action = 3
608     SKIP
609     IF EOF()
610         ?? CHR( 7 )
611         WAIT "Last record" WINDOW NOMAIT
612         SKIP -1
613     ENDIF
614 CASE m.action = 4
615     SKIP -1
616     IF BOF()
617         ?? CHR( 7 )
618         WAIT "First record" WINDOW NOMAIT
619         SKIP
620     ENDIF
621 CASE m.action = 5
622     GO TOP
623 CASE m.action = 6
624     GO BOTTOM
625 CASE m.action = 7
626     m.item = browseitem(0)
627     IF m.item=0
628         LOCATE FOR hmcfeid =m.item
629         IF FOUND()
630             GO RECD()
631             SCATTER MEMVAR
632             IF m.hmcfeid>0
633                 m.answr=get_hmcfe(m.hmcfeid)
634             ELSE
635                 m.answr=""
636             ENDIF
637             IF m.hmcfeid>0
638                 m.hmcfe=get_hmcfe(m.hmcfeid)
639             ELSE
640                 m.hmcfe=""
641             ENDIF
642             m.oldhmcfe=m.hmcfe
643             m.oldanswr=m.answr
644             SHOW GETS
645         ENDIF
646     ENDIF
```


* _QKFVRL1J m.Save VALID

Function Origin:

From Platform: MS-DOS
From Screen: HMCFEI, Record Number: 12
Variable: m.Save
Called By: VALID Clause
Object Type: Push Button
Snippet Number: 10

* * did the record change
FUNCTION _qkfvr1j && m.Save VALID

#REGION 1_qkfvr1j && m.Save VALID
DO CHANGE

IF m.save = 1 && Selected Save Button

IF m.adding && Adding a new record
APPEND BLANK
GATHER MEMVAR
ENDIF

IF m.change && Changing an old record
GATHER MEMVAR
ENDIF

SCATTER MEMVAR

IF m.hmcfid>0
m.answr=get_hmcfe(m.hmcfid)
ELSE
m.hmcfe=""
ENDIF

IF m.hmcfeid>0
m.hmcfe=get_hmcfe(m.hmcfeid)
ELSE
m.hmcfe=""
ENDIF

m.oldhmcfe=m.hmcfe
m.oldanswr=m.answr
SHOW GETS
SHOW GET action enabled
SHOW GET m.factor enabled
SHOW GET m.element disabled
SHOW GET SAVE disabled
SHOW GET m.hmcfe disabled
SHOW GET m.hmcfeid disabled
SHOW GET m.hmcfeino disabled
SHOW GET m.hmcfeicost disabled
SHOW GET m.answr disabled

m.adding=.f.
m.change=.f.
m.editing = .f.
*: EOF: HMCFEI.AC1


```

132 SIZE 1,10,0
133 a 1,25 SAY "Scenario:";
134 SIZE 1,9,0
135 a 17,21 SAY "Total of Scenario cost:";
136 SIZE 1,23,0
137 a 16,1 TO 19,19
138 a 0,1 SAY "Estimated";
139 SIZE 1,9,0
140 a 1,2 SAY "Cost";
141 SIZE 1,4,0
142 a 0,64 SAY "Estimated";
143 SIZE 1,9,0
144 a 1,65 SAY "Variance";
145 SIZE 1,8,0
146 a 17,3 SAY "of Iterations";
147 SIZE 1,15,0
148 a 1,34 GET m.chmscid;
149 SIZE 1,3,0
150 DEFAULT 0;
151 PICTURE "a2";
152 DISABLE
153 a 1,38 GET m.chmscname;
154 SIZE 1,15,0
155 DEFAULT " ";
156 PICTURE "a1";
157 DISABLE
158 a 3,8 SAY "for Product:";
159 SIZE 1,12,0
160 a 2,21 GET m.hmat;
161 PICTURE "a3";
162 FROM hmatarray;
163 SIZE 3,42;
164 DEFAULT 1;
165 COLOR SCHEME 1, 2
166 a 5,21 GET m.step;
167 PICTURE "a4";
168 FROM steparray;
169 SIZE 3,42;
170 DEFAULT 1;
171 COLOR SCHEME 1, 2
172 a 6,2 GET m.cstep;
173 PICTURE "a5";
174 SIZE 1,5,0;
175 DEFAULT 0;
176 a 6,67 GET m.bstep;
177 PICTURE "a6";
178 SIZE 1,5,0;
179 DEFAULT 0;
180 VALID qle016b8t()
181 a 8,21 GET m.fact;
182 PICTURE "a7";
183 FROM factarray;
184 SIZE 3,42;
185 DEFAULT 1;
186 COLOR SCHEME 1, 2
187 a 9,2 GET m.cfact;
188 PICTURE "a8";
189 SIZE 1,5,0;
190 DEFAULT 0;
191 a 9,67 GET m.bfact;
192 PICTURE "a9";
193 SIZE 1,5,0;
194 DEFAULT 0;
195 VALID qle016biv()
196 a 11,21 GET m.wfact;
197 PICTURE "a10";

```

```

198 FROM factarray;
199 SIZE 3,19;
200 DEFAULT 1;
201 COLOR SCHEME 1, 2
202 a 11,48 GET m.wstep;
203 PICTURE "a11";
204 FROM steparray;
205 SIZE 3,15;
206 DEFAULT 1;
207 COLOR SCHEME 1, 2
208 a 12,2 GET m.cwstep;
209 PICTURE "a12";
210 SIZE 1,5,0;
211 DEFAULT 0;
212 a 12,67 GET m.bwstep;
213 PICTURE "a13";
214 SIZE 1,5,0;
215 DEFAULT 0;
216 VALID qle016br7()
217 a 14,21 GET m.bphase;
218 PICTURE "a14";
219 FROM phasearray;
220 SIZE 3,42;
221 DEFAULT 1;
222 COLOR SCHEME 1, 2
223 a 15,2 GET m.cphase;
224 PICTURE "a15";
225 SIZE 1,5,0;
226 DEFAULT 0;
227 a 15,67 GET m.bphase;
228 PICTURE "a16";
229 SIZE 1,5,0;
230 DEFAULT 0;
231 VALID qle016bvm()
232 a 17,47 GET m.scost;
233 PICTURE "a17" Yes ;No";
234 SIZE 1,10,0;
235 DEFAULT 2;
236 a 18,3 GET m.sample;
237 SIZE 1,12;
238 DEFAULT 0;
239 PICTURE "a18";
240 VALID qle016c5n()
241 a 19,24 GET m.action;
242 PICTURE "a19" \<OK;\<Browse;\<Cancel";
243 SIZE 1,8,3;
244 DEFAULT 1;
245 VALID qle016cb0()
246
247 [IF NOT WVISIBLE("w_hmcost")
248 ACTIVATE WINDOW w_hmcost
249 ENDIF
250
251 READ CYCLE MODAL
252
253 RELEASE WINDOW w_hmcost
254
255 #REGION 0
256 [IF m.talkstat = "ON"
257 SET TALK ON
258 ENDIF
259 [IF m.comstat = "ON"
260 SET COMPATIBLE ON
261 ENDIF
262
263

```

Line	Code	Line	Code
264	*	330	
265	*	331	
266	*	332	
267	*	333	
268	*	334	
269	*	335	
270	*	336	
271		337	
272	#REGION 1	338	
273	=delete("hmcamp")	339	
274	=delete("hmtamp")	340	
275	=delete("hmtest")	341	
276	=delete("hmtmp")	342	
277	SET SAFETY &unsafe	343	
278	ON ERROR &moderr	344	
279	RETURN	345	
280	*****	346	
281		347	
282	*	348	
283	*	349	
284	*	350	
285	*	351	
286	*	352	
287	*	353	
288		354	
289		355	
290		356	
291		357	
292	#REGION 1	358	
293	PROCEDURE OPEN	359	
294	*****	360	
295		361	
296	filesuccess = openfile("hmstep")	362	
297	DO CANCEL WITH filesuccess	363	
298		364	
299	filesuccess = openfile("hmtab")	365	
300	DO CANCEL WITH filesuccess	366	
301	DO CANCEL WITH filesuccess	367	
302	SET ORDER TO TAG hmatid	368	
303		369	
304	filesuccess = openfile("hmlc")	370	
305	DO CANCEL WITH filesuccess	371	
306	SET ORDER TO TAG hmlcid	372	
307		373	
308	filesuccess = openfile("hmp")	374	
309	DO CANCEL WITH filesuccess	375	
310	SET ORDER TO TAG hmpid	376	
311		377	
312	filesuccess = openfile("hmc")	378	
313	DO CANCEL WITH filesuccess	379	
314	SET ORDER TO TAG hmcid	380	
315		381	
316	RETURN	382	
317	*****	383	
318	FUNCTION delete	384	
319	*****	385	
320	PARAMETER FILE	386	
321	IF USED(FILE)	387	
322	SELECT (FILE)	388	
323	USE	389	
324	DELETE FILE (FILE + ".dbf")	390	
325	END IF	391	
326	RETURN	392	
327	*****	393	
328	PROCEDURE CANCEL	394	
329	*****	395	

```

330  PARAMETER success
331  IF lm_success
332    =errmsg("file doesn't Exist !!!", 2)
333    CANCEL
334    ENDIF
335    RETURN
336
337  *****
338  PROCEDURE get_hmarray
339  *****
340  PRIVATE msel
341  msel = SELECT()
342  SELECT *
343    FROM hmstep;
344    WHERE hmscid = chmscid;
345    INTO TABLE hmtemp
346
347
348  SELECT hmtemp
349  SET RELATION TO hmlcid INTO hmlc
350  IF RECOUNT() > 0
351    DECLARE steparray[RECOUNT(), 2]
352    i = 0
353    j = 0
354    SCAN
355      i = i + 1
356      steparray[i,1] = STR(hmstep)
357      steparray[i,2] = RECNO()
358    DO get_table WITH hmlcid, hmlc, hmscid
359      m.lcphase = ALLTRIM(hmlc.hmlc)
360      IF ASCAN(phasearray, lcphase) = 0
361        j = j + 1
362        DECLARE phasearray[j,2]
363        phasearray[j,1] = ALLTRIM(hmlc.hmlc)
364        phasearray[j,2] = hmlcid
365      ENDIF
366    ENDSCAN
367  IF i > 1
368    DECLARE steparray[i+1,2]
369    =AINS(steparray,1)
370    steparray[i,1] = "All Steps"
371    steparray[i,2] = 0
372  ENDIF
373
374  IF j > 1
375    DECLARE phasearray[j + 1,2]
376    =AINS(phasearray,1)
377    phasearray[1,1] = "All phases"
378    phasearray[1,2] = 0
379  ENDIF
380  SET RELATION TO      && close relationship
381
382  SELECT hmcomp
383  IF RECOUNT() > 0
384    * Create array for factor
385    SET RELATION TO hmcfid INTO hmcf
386    i = 0
387    SCAN
388      factor = ALLTRIM(hmcf.hmcf)
389      IF ASCAN(factorarray, factor) = 0
390        i = i + 1
391        DECLARE factarray[i, 2]
392        factarray[i,1] = ALLTRIM(hmcf.hmcf)
393        factarray[i,2] = hmcfid
394      ENDIF
395    ENDSCAN
396    ASORT = (factarray)

```

```

396 IF i > 1
397   DECLARE factarray[i+1,2]
398   = AINS(factarray,1)
399   factarray[1,1] = "All factors"
400   factarray[1,2] = 0
401   ENDIF
402
403 * Create array for material
404
405 SET RELATION TO hmatid INTO hmat
406 i = 0
407
408 SCAN
409   material = ALLTRIM(hmat.hmatname) + IIF(EMPTY(hmat.mfg),"","")
410   + ALLTRIM(hmat.mfg) + " "
411   IF ASCAN(hmatarray, material) = 0
412     DECLARE hmatarray[i, 2]
413     hmatarray[i,1] = ALLTRIM(hmat.hmatname) + IIF(EMPTY(hmat.
414     mfg) ,"", " "
415     + ALLTRIM(hmat.mfg) + " ")
416     hmatarray[i,2] = hmatid
417     ENDIF
418   ASORT = (hmatarray)
419   IF i > 1
420     DECLARE hmatarray[i+1,2]
421     = AINS(hmatarray,1)
422     hmatarray[1,1] = "All products"
423     hmatarray[1,2] = 0
424     ENDIF
425   SELECT (mset)
426   RETURN
427
428 *****
429 PROCEDURE get table
430 *****
431 PARAMETER comid, lcld, wpid
432
433 PRIVATE mset
434 mset = SELECT()
435 SELECT *
436 FROM hmtab;
437 WHERE hmcomid = m.comid AND hmlcid = m.lcid AND hmwpid = m.wpid;
438 INTO TABLE temp
439
440 SELECT hmcoid
441 APPEND FROM temp
442 SELECT (mset)
443
444 RETURN
445
446 PROCEDURE compute
447 *****
448 PUBLIC asize
449 IF hmatarray[m.hmat,2] = 0
450   asize = ALEN(hmatarray,1)
451   FOR INDEX = 2 TO asize
452     = subcomput(ALLTRIM(hmatarray[index,1]), hmatarray[index,2])
453   ENDFOR
454 ELSE
455   = subcomput(ALLTRIM(hmatarray[m.hmat,1]), hmatarray[m.hmat,2])
456 ENDIF
457 DO DISPLAY WITH m.finalstr
458 RETURN
459

```

```

460 *****
461 PROCEDURE subcomput
462 *****
463 PARAMETER m.ihmatname, m.ihmatid
464
465 PRIVATE mset
466 mset = SELECT()
467 scntotal = 0
468 DIMENSION steptotal[1], hmattotal[1], facttotal[1], wfacttotal[1], phas
469 => etotal[1]
470
471 IF m.cstep = 1
472   IF steptarray[m.step,2] = 0
473     SELECT hmtemp
474     GO TOP
475     DIMENSION steptotal[RECCOUNT()]
476     i = 0
477     SCAN
478       i = i + 1
479       m.stotal = computstep(hmtemp.hmcoid, m.ihmatid, hmtemp.hmlc
480       => id, hmtemp.hmwpid, hmtemp.pernum, hmtemp.durnum, hmtemp.qtnum)
481       steptotal[i] = ALLTRIM(STR(hmstep)) + u + ALLTRIM(STR(m.stot
482       => al,8,2))
483       scntotal = scntotal + m.stotal
484     ENDSCAN
485   ELSE
486     DIMENSION steptotal[1]
487     SELECT hmtemp
488     GO steptarray[m.step,2]
489     m.stotal = computstep(hmtemp.hmcoid, m.ihmatid, hmtemp.hmlcid,
490     => hmtemp.hmwpid, hmtemp.pernum, hmtemp.durnum, hmtemp.qtnum)
491     steptotal[i] = ALLTRIM(STR(hmtemp.hmstep)) + u + ALLTRIM(STR(m.
492     => stotal,8,2))
493   ENDIF
494
495 IF m.scost = 1
496   IF EMPTY(scntotal)
497     SELECT hmtemp
498     GO TOP
499     SCATTER MEMVAR
500     m.stotal = computstep(m.hmcoid, m.ihmatid, m.hmlcid, m.hmwpid
501     => id, m.pernum, m.durnum, m.qtnum)
502     scntotal = scntotal + m.stotal
503   ENDSCAN
504   ENDIF
505
506 IF m.cfact = 1
507   IF factarray[m.fact,2] = 0
508     SIZE = ALEN(factarray,1)
509     DIMENSION facttotal[SIZE - 1]
510     FOR j = 2 TO SIZE
511       m.stotal = computfact(factarray[j,2], m.ihmatid)
512       facttotal[j - 1] = ALLTRIM(factarray[j,1]) + u + ALLTRIM(STR
513       => (m.stotal,8,2))
514     ENDFOR
515   ELSE
516     DIMENSION facttotal[1]
517     m.stotal = computfact(factarray[m.fact,2], m.ihmatid)
518     facttotal = ALLTRIM(factarray[m.fact,1]) + u + ALLTRIM(STR(m.sto
519     => tal,8,2))
520   ENDIF
521 ENDIF

```

UNCOMP.AC1 12-1-93 11:32a

```

632 IF PARAMETER() = 0
633 RETURN 0
634 ENDIF
635 msel = SELECT()
636 SELECT hntemp
637 GO TOP
638 m.sum = 0
639
640 m.wpid = hntemp.hmwpid
641 m.comid = hntemp.hmcomid
642 m.lcid = hntemp.hmlcid
643 SELECT SUM(hmcomp.wtaverage), 0000.00, hmcomp.hmetid, hmcomp.hmetpr
644 => ob, hmcomp.hmcfeid, hmcomp.hmcfeid, hmcomp.perp, hmcomp.perq, hmcomp.
645
646 FROM hmcomp;
647 WHERE hmcomp.hmcomid = m.comid;
648 AND hmcomp.hmetid = m.hmetid;
649 AND hmcomp.hmlcid = m.lcid;
650 AND hmcomp.hmwpid = m.wpid;
651 AND hmcomp.hmcfeid = m.factid;
652 GROUP BY hmcomp.hmetid, hmcomp.hmcfeid, hmcomp.hmcfeid;
653 ORDER BY hmcomp.hmetid;
654 INTO TABLE test
655 m.sum = m.sum + CALCULATE(hntemp.pernum, hntemp.durnum, hntemp.qty
656
657 ENDSCAN
658 SELECT (msel)
659 RETURN m.sum
660
661 *****
662 FUNCTION computefact
663 *****
664 PARAMETER m.factid, m.hmetid
665 PRIVATE msel
666 IF PARAMETER() = 0
667 RETURN 0
668 ENDIF
669 msel = SELECT()
670 IF steparray(m.wstep, 2) = 0
671 SELECT hntemp
672 m.resul = ""
673 GO TOP
674
675 m.comid = hntemp.hmcomid
676 m.lcid = hntemp.hmlcid
677 m.wpid = hntemp.hmwpid
678 SELECT SUM(hmcomp.wtaverage), 0000.00, hmcomp.hmetid, hmcomp.hme
679 => tprob, hmcomp.hmcfeid, hmcomp.hmcfeid, hmcomp.perp, hmcomp.perq, hmco
680 => mp.hmunit;
681 FROM hmcomp;
682 WHERE hmcomp.hmcomid = m.comid;
683 AND hmcomp.hmetid = m.hmetid;
684 AND hmcomp.hmlcid = m.lcid;
685 AND hmcomp.hmwpid = m.wpid;
686 AND hmcomp.hmcfeid = m.factid;
687 GROUP BY hmcomp.hmetid, hmcomp.hmcfeid, hmcomp.hmcfeid;
688 ORDER BY hmcomp.hmetid;
689 INTO TABLE test
690 m.resul = IF(EMPTY(m.resul), "", m.resul + u) + ALLTRIM(STR(hm
691 => temp.hmetpr) + " " + ALLTRIM(STR(CALCULATE(hntemp.pernum, hntemp.durnum, h
692 => mtemp.qtyynum), 8, 2))
693 ENDSCAN
694 ELSE

```

```

691 SELECT hntemp
692 GO (steparray(m.wstep, 2))
693 m.comid = hntemp.hmcomid
694 m.lcid = hntemp.hmlcid
695 m.wpid = hntemp.hmwpid
696 SELECT SUM(hmcomp.wtaverage), 0000.00, hmcomp.hmetid, hmcomp.hmetpr
697 => ob, hmcomp.hmcfeid, hmcomp.hmcfeid, hmcomp.perp, hmcomp.perq, hmcomp.
698 => hmunit;
699 FROM hmcomp;
700 WHERE hmcomp.hmcomid = m.comid;
701 AND hmcomp.hmetid = m.hmetid;
702 AND hmcomp.hmlcid = m.lcid;
703 AND hmcomp.hmwpid = m.wpid;
704 AND hmcomp.hmcfeid = m.factid;
705 GROUP BY hmcomp.hmetid, hmcomp.hmcfeid, hmcomp.hmcfeid;
706 ORDER BY hmcomp.hmetid;
707 INTO TABLE test
708 m.resul = ALLTRIM(STR(hntemp.hmetpr) + " " + ALLTRIM(STR(CALCULAT
709 => E(hntemp.pernum, hntemp.durnum, hntemp.qtyynum), 8, 2))
710 ENDIF
711 RETURN (msel)
712
713 *****
714 FUNCTION computephase
715 *****
716 PARAMETER m.phaseid, m.hmetid
717 PRIVATE m.sum, msel
718 IF PARAMETER() = 0
719 RETURN 0
720 ENDIF
721 msel = SELECT()
722 SELECT hntemp
723 GO TOP
724 m.sum = 0
725
726 IF hmlcid = m.phaseid
727 m.comid = hntemp.hmcomid
728 m.lcid = hntemp.hmlcid
729 m.wpid = hntemp.hmwpid
730 SELECT SUM(hmcomp.wtaverage), 0000.00, hmcomp.hmetid, hmcomp.hme
731 => tprob, hmcomp.hmcfeid, hmcomp.hmcfeid, hmcomp.perp, hmcomp.perq, hmco
732 => mp.hmunit;
733 FROM hmcomp;
734 WHERE hmcomp.hmcomid = m.comid;
735 AND hmcomp.hmetid = m.hmetid;
736 AND hmcomp.hmlcid = m.lcid;
737 AND hmcomp.hmwpid = m.wpid;
738 GROUP BY hmcomp.hmetid, hmcomp.hmcfeid, hmcomp.hmcfeid;
739 ORDER BY hmcomp.hmetid;
740 INTO TABLE test
741 m.sum = m.sum + CALCULATE(hntemp.pernum, hntemp.durnum, hntemp.
742 => qtyynum)
743 ENDIF
744 ENDSCAN
745 SELECT (msel)
746 RETURN m.sum
747
748 *****
749 FUNCTION CALCULATE
750 *****
751 PARAMETER sn, sd, sq
752 PRIVATE msel

```

```

751 msel = SELECT()
752 IF IUSED("test")
753 RETURN 0
754 ENDIF
755 SELECT test
756 t = 0
757 IF RECCOUNT() > 0
758 SCAN
759 s = 1
760 IF perp = 1
761 s = s * sn
762 ENDIF
763 IF perp = 1
764 s = s * sd
765 ENDIF
766 IF perp = 1
767 s = s * sq
768 ENDIF
769 s = sum wtaver * s
770 REPLACE exp_2 WITH s
771 ENDSCAN
772 ENDIF
773
774 SELECT SUM(exp_2), hmetprob;
775 FROM test;
776 GROUP BY hmetid;
777 INTO CURSOR EXP;
778
779 SELECT EXP
780 IF RECCOUNT() > 0
781 t = 0
782 SCAN
783 IF EMPTY(hmetprob)
784 hmetprob = 11F(RECNO())=1,1,.01)
785 ENDIF
786 t = t + (sum_exp_2 * hmetprob)
787 ENDSCAN
788
789 =del file('EXP')
790 SELECT (msel)
791 RETURN t
792
793 *****
794 FUNCTION resetwt
795 *****
796 PARAMETER matid, lcpfase, wp
797 msel = SELECT()
798 SELECT test
799 SCAN
800 m.wt = selwt(m.matid, m.lcpfase, m.wp, test.hmetid, test.hmcfid, t
=> est.hmcfid)
801 REPLACE sum_wtaver WITH m.wt
802 ENDSCAN
803 SELECT (msel)
804 RETURN
805
806 *****
807 FUNCTION selwt
808 *****
809 PARAMETER matid, lcpfase, wp, metid, cfid, cfeid
810 msel = SELECT()
811 SELECT hmcfcost, prob;
812 FROM hmtab;
813 WHERE hmtab.hmetid = m.matid;
814 AND hmtab.hmcid = m.lcpfase;
815 AND hmtab.hmwpid = m.wp;

```

```

816 AND hmtab.hmetid = m.matid;
817 AND hmtab.hmcid = m.cfid;
818 AND hmtab.hmcfeid = m.cfeid;
819 INTO TABLE test1
820
821 SELECT test1
822 m.random = RAND()
823 m.wt = 0
824 m.start = 0
825 m.end = 0
826
827 SCAN
828 m.end = 11F(EMPTY(m.end), test1.prob, END + test1.prob)
829 IF m.random > m.start AND m.random < m.end
830 m.wt = test1.hmcfcost
831 ENDIF
832 m.start = m.start + test1.prob
833 ENDSCAN
834 =del file('test1')
835 SELECT (msel)
836 RETURN m.wt
837
838 *****
839 FUNCTION setupboot
840 *****
841
842 CASE m.bstep = 1
843 IF stepparray[m.step,2] = 0
844 =ACOPY(stepparray, temparray)
845 =ADEL(temparray, 1)
846 DECLARE temparray(ALEN(temparray,1) - 1,2)
847 m.temp = bselect(temparray, "Select step: ")
848 IF EMPTY(m.temp)
849 =errmsg("No Bootstrap computing...",2)
850 m.bstep = 0
851 ELSE
852 m.pos = ASCAN(m.stepparray, m.temp)
853 m.bstep = ASUBSCRIPT(m.stepparray, m.pos, 1)
854 ENDIF
855 RELEASE temparray
856 ELSE
857 m.bstep = m.step
858 ENDIF
859
860 CASE m.bfact = 1
861 IF factarray[m.fact,2] = 0
862 =ACOPY(factarray, temparray)
863 =ADEL(temparray, 1)
864 DECLARE temparray(ALEN(temparray,1) - 1,2)
865 m.temp = bselect(temparray, "Select factor: ")
866 IF EMPTY(m.temp)
867 =errmsg("No Bootstrap computing...",2)
868 m.bfact = 0
869 ELSE
870 m.pos = ASCAN(m.factarray, m.temp)
871 m.bfact = ASUBSCRIPT(m.factarray, m.pos, 1)
872 ENDIF
873 RELEASE temparray
874 ELSE
875 m.bfact = m.fact
876 ENDIF
877
878 CASE m.bwstep = 1
879 IF stepparray[m.wstep,2] = 0 OR factarray[m.wfact,2] = 0
880 =ACOPY(factarray, temparray)
881 IF temparray[1,2] = 0

```

```

882      =ADEL(temparray,1)
883      DECLARE temparray(ALEN(temparray,1) - 1,2)
884      ENDIF
885      =ACOPY(steparray, temparray)
886      IF temparray[1,2] = 0
887        =ADEL(temparray,1)
888        DECLARE temparray1(ALEN(temparray,1) - 1,2)
889        ENDIF
890        m.temp = bfact(atemparray,atemparray)
891        IF EMPTY(m.temp)
892          =errmsg("No Bootstrap computing...",2)
893          m.bwstep = 0
894        ELSE
895          m.pos = ASCAN(factarray,do(m.temp,u,1))
896          m.bbwfact = ASUBSCRIPT(m.factarray,m.pos,1)
897          m.pos = ASCAN(steparray,do(m.temp,u,2))
898          m.bbwstep = ASUBSCRIPT(m.steparray,m.pos,1)
899          ENDIF
900          RELEASE temparray, temparray1
901        ELSE
902          m.bbwstep = m.step
903          m.bbwfact = m.fact
904        ENDIF
905      CASE m.bphase = 1
906      IF phasearray[m.phase,2] = 0
907        =ACOPY(phasearray, temparray)
908        =ADEL(temparray,1)
909        DECLARE temparray1(ALEN(temparray,1) - 1,2)
910        m.temp = bselect(atemparray,"Select Life cycle phase:")
911        IF EMPTY(m.temp)
912          =errmsg("No Bootstrap computing...",2)
913          m.bphase = 0
914        ELSE
915          m.pos = ASCAN(phasearray, m.temp)
916          m.bbwphase = ASUBSCRIPT(m.phasearray,m.pos,1)
917          ENDIF
918        ELSE
919          m.bbwphase = m.phase
920        ENDIF
921      ENDCASE
922      SHOW GETS
923      RETURN
924
925      *****
926      FUNCTION bstep
927      *****
928      PARAMETER mat, lcpbase, wp, sn, sd, sq
929      CREAT TABLE boot (cost N(10,2))
930      m.boot = ""
931      SELECT SUM(hmcomp.wtaverage), 0000.00, hmcomp.hmetid, hmcomp.hmetprob
932      => hmcomp.hmcfeid,
933      hmcomp.hmcfeid, hmcomp.perp, hmcomp.perd, hmcomp.perq, hmcomp.hmu
934      => nit;
935      FROM hmcomp;
936      WHERE hmcomp.hmetid = m.mat;
937      AND hmcomp.hmcid = m.lcpbase;
938      AND hmcomp.hmwpid = m.wp;
939      GROUP BY hmcomp.hmetid, hmcomp.hmcfeid, hmcomp.hmcfeid;
940      ORDER BY hmcomp.hmetid;
941      INTO TABLE test
942      FOR i = 1 TO m.sample
943      = resetwt(m.mat, m.lcpbase, m.wp)
944      m.sum = CALCULATE(sn, sd, sq)
945      SELECT boot

```

```

946      APPEND BLANK
947      REPLACE cost WITH m.sum
948      ENDFOR
949      CALCULATE STD(cost), AVG(cost) TO m.std, m.mean
950      m.boot = ALLTRIM(STR(m.std,8,4)) + u + ALLTRIM(STR(m.mean,8,2))
951      =delete('boot')
952      RETURN m.boot
953
954      *****
955      FUNCTION bfact
956      *****
957      PARAMETER m.factid, m.hmetid
958      PRIVATE msel
959      IF PARAMETER() = 0
960        RETURN 0
961      ENDIF
962      m.boot = ""
963      msel = SELECT()
964      CREATE TABLE boot (cost N(10,2))
965      FOR i = 1 TO m.sample
966        SELECT hmtemp
967        GO TOP
968        m.sum = 0
969        SCAN
970          m.wpid = hmtemp.hmwpid
971          m.cmid = hmtemp.hmcid
972          m.lcid = hmtemp.hmlcid
973          SELECT SUM(hmcomp.wtaverage), 0000.00, hmcomp.hmetid, hmcomp.perd,
974          hmcomp.hmcfeid, hmcomp.hmcfeid, hmcomp.perp, hmcomp.perq,
975          hmcomp.hmu;
976          FROM hmcomp;
977          WHERE hmcomp.hmcid = m.cmid;
978          AND hmcomp.hmetid = m.hmetid;
979          AND hmcomp.hmwpid = m.lcid;
980          AND hmcomp.hmcfeid = m.wpid;
981          GROUP BY hmcomp.hmetid, hmcomp.hmcfeid, hmcomp.hmcfeid;
982          ORDER BY hmcomp.hmetid;
983          INTO TABLE test
984          =resetwt(m.hmetid, m.lcid, m.wpid)
985          m.sum = m.sum + CALCULATE(hmtemp.perp, hmtemp.perq, hmtemp.hmu)
986          => qty(m)
987      ENDSCAN
988      SELECT boot
989      APPEND BLANK
990      REPLACE cost WITH m.sum
991      ENDFOR
992      SELECT boot
993      CALCULATE STD(cost), AVG(cost) TO m.std, m.mean
994      m.boot = ALLTRIM(STR(m.std,8,4)) + u + ALLTRIM(STR(m.mean,8,2))
995      =delete('boot')
996      SELECT (msel)
997      RETURN m.boot
998
999      *****
1000      FUNCTION bwstep
1001      *****
1002      PARAMETER m.factid, m.stepid, m.hmetid
1003      PRIVATE msel
1004      IF PARAMETER() = 0
1005        RETURN 0
1006      ENDIF
1007      m.boot = ""
1008      msel = SELECT()
1009      CREATE TABLE boot (cost N(10,2))

```



```

1010 SELECT hntemp
1011 GO (m.stepid)
1012 m.comid = hntemp.hmc.comid
1013 m.lcid = hntemp.hmlcid
1014 m.wpid = hntemp.hmwpid
1015 SELECT SUM(hntemp.waverage), 0000.00, hntemp.hmetid, hntemp.hmetprob,
1016 hntemp.hmcfeid, hntemp.perp, hntemp.perd, hntemp.perq, hntemp.hmu
1017 => nit;
1018 FROM hntemp;
1019 WHERE hntemp.hmc.comid = m.comid;
1020 AND hntemp.hmetid = m.hmetid;
1021 AND hntemp.hmlcid = m.lcid;
1022 AND hntemp.hmwpid = m.wpid;
1023 AND hntemp.hmcfeid = m.hmcfeid;
1024 GROUP BY hntemp.hmetid, hntemp.hmcfeid, hntemp.hmcfeid;
1025 ORDER BY hntemp.hmetid;
1026 INTO TABLE test
1027 FOR i = 1 TO m.sample
1028 =reset(m.hmetid, m.lcid, m.wpid)
1029 m.sum = CALCULATE(hntemp.pernum, hntemp.durnum, hntemp.qtnum)
1030 SELECT boot
1031 APPEND BLANK
1032 REPLACE cost WITH m.sum
1033 ENDFOR
1034 SELECT boot
1035 CALCULATE STD(cost), AVG(cost) TO m.std, m.mean
1036 m.boot = ALLTRIM(STR(m.std,8,4)) + u + ALLTRIM(STR(m.mean,8,2))
1037 =delete('boot')
1038 SELECT (msel)
1039 RETURN m.boot
1040 *****
1041 FUNCTION bphase
1042 *****
1043 PARAMETER m.phaseid, m.hmetid
1044 PRIVATE m.sum, msel
1045 IF PARAMETER() = 0
1046 RETURN 0
1047 ENDIF
1048 m.boot = ""
1049 msel = SELECT()
1050 CREATE TABLE boot (cost N(10,2))
1051 FOR i = 1 TO m.sample
1052 SELECT hntemp
1053 GO TOP
1054 m.sum = 0
1055 SCAN
1056 IF hmlcid = m.phaseid
1057 m.comid = hntemp.hmc.comid
1058 m.lcid = hntemp.hmlcid
1059 m.wpid = hntemp.hmwpid
1060 SELECT SUM(hntemp.waverage), 0000.00, hntemp.hmetid, hntemp.
1061 hntemp.hmcfeid, hntemp.perp, hntemp.perd, hntemp.perq, h
1062 => hntemp.hmu;
1063 FROM hntemp;
1064 WHERE hntemp.hmc.comid = m.comid;
1065 AND hntemp.hmetid = m.hmetid;
1066 AND hntemp.hmlcid = m.lcid;
1067 AND hntemp.hmwpid = m.wpid;
1068 GROUP BY hntemp.hmetid, hntemp.hmcfeid, hntemp.hmcfeid;
1069 ORDER BY hntemp.hmetid;
1070 *****
1071 INTO TABLE test
1072 =reset(m.hmetid, m.lcid, m.wpid)
1073 m.sum = m.sum + CALCULATE(hntemp.pernum, hntemp.durnum, hntemp.
1074 hntemp.hmcfeid, hntemp.perp, hntemp.perd, hntemp.perq, hntemp.hmu)
1075 ENDIF
1076 ENDSCAN
1077 SELECT boot
1078 APPEND BLANK
1079 REPLACE cost WITH m.sum
1080 ENDFOR
1081 SELECT boot
1082 CALCULATE STD(cost), AVG(cost) TO m.std, m.mean
1083 m.boot = ALLTRIM(STR(m.std,8,4)) + u + ALLTRIM(STR(m.mean,8,2))
1084 =delete('boot')
1085 SELECT (msel)
1086 RETURN m.boot
1087 *****
1088 FUNCTION getstr
1089 *****
1090 PARAMETERS hmetname, scentotal, steptotal, factotal, wfactotal, phase
1091 => total, bootstrap
1092 EXTERNAL ARRAY steptotal, factotal, wfactotal, phasetotal
1093 m.text = ""
1094 u = ""
1095 m.indent = SPACE(5)
1096 m.text = hmetname + ":" + newline
1097 IF EMPTY(scentotal)
1098 m.text = m.text + m.indent + "Total Cost of scenario" + ":" + " +
1099 STR(scentotal,8,2) + newline
1100 m.text = m.text + newline
1101 ENDIF
1102 IF EMPTY(steptotal)
1103 FOR i = 1 TO ALEN(steptotal)
1104 m.text = m.text + m.indent + "Cost of STEP " + dp(steptotal[i],u
1105 => ,1) + ":" + dp(steptotal[i],u,2) + newline
1106 ENDFOR
1107 m.text = m.text + newline
1108 ENDIF
1109 IF EMPTY(factotal)
1110 FOR i = 1 TO ALEN(factotal)
1111 m.text = m.text + m.indent + "Cost of FACTOR " + dp(factotal[i],
1112 => ,1) + ":" + dp(factotal[i],u,2) + newline
1113 ENDFOR
1114 m.text = m.text + newline
1115 ENDIF
1116 IF EMPTY(wfactotal)
1117 FOR i = 1 TO ALEN(wfactotal)
1118 m.wd = wfactotal[i]
1119 IF EMPTY(m.wd)
1120 m.fid = dp(m.wd,u,1)
1121 m.sd = dp(m.wd,u,2,999)
1122 DO WHILE EMPTY(m.sd)
1123 m.dd = dp(m.sd,u,1)
1124 IF OCCURS(u, m.sd) > 0
1125 m.sd = dp(m.sd,u,2,999)
1126 ELSE
1127 m.sd = ""
1128 ENDIF
1129 m.text = m.text + m.indent + "Cost of FACTOR " + m.fid + " +
1130 => within STEP " + dp(m.dd,"",1) + ":" + dp(m.dd,"",2) + newline
1131 ENDDO
1132 ENDIF
1133 ENDFOR
1134 m.text = m.text + newline

```

```

1132 _ENDIF
1133 IF EMPTY(phasetotal)
1134 FOR i = 1 TO ALEN(phasetotal)
1135   m.text = m.text + mindent + "Cost of PHASE " + dp(phasetotal[i])
1136   + ", "
1137   + " " + dp(phasetotal[i], 2) + newline
1138 _ENDIF
1139 m.text = m.text + newline
1140 IF EMPTY(bootstrap)
1141   m.text = m.text + mindent + bootstrap + newline
1142 _ENDIF
1143 RETURN m.text

```

```

1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159

```

```

1160 FUNCTION _qle016b8t  && m.bstep VALID

```

```

1161 #REGION 1

```

```

1162 IF bstep = 1
1163   bfact = 0
1164   bstep = 0
1165   bphase = 0
1166   SHOW GETS
1167 _ENDIF

```

```

1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196

```

```

1197 FUNCTION _qle016br7  && m.bstep VALID

```

```

1198 #REGION 1
1199 IF bstep = 1
1200   bfact = 0
1201   bstep = 0
1202   bphase = 0
1203   SHOW GETS
1204 _ENDIF

```

```

1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262

```

```

1263 FUNCTION _qle016biv  && m.bfact VALID

```

```

1264 #REGION 1
1265 IF bfact = 1
1266   bstep = 0
1267   bphase = 0
1268   SHOW GETS
1269 _ENDIF

```

```

1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000

```

```

1299 FUNCTION _qle016br7  && m.bstep VALID

```

```

1300 #REGION 1
1301 IF bstep = 1
1302   bfact = 0
1303   bstep = 0
1304   bphase = 0
1305   SHOW GETS
1306 _ENDIF

```

```

1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
```

```

1263 RETURN .F.
1264 ENDIF
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296

```

_OLE016CB0	m.action VALID
Function Origin:	
From Platform:	MS-DOS
From Screen:	HMCOMP,
Variable:	m.action
Called By:	VALID Clause
Object Type:	Push Button
Snippet Number:	6
	Record Number: 34

```

1281 FUNCTION _ole016cb0  && m.action VALID
1282 #REGION 1
1283 DO CASE
1284 CASE m.action = 1
1285     =setupboot()
1286     wpop = popupshow("Calculating....")
1287     DO compute
1288     =popuphide(wpop)
1289     CLEAR READ
1290 CASE m.action = 2
1291     DO hmbrowse
1292 CASE m.action = 3
1293     CLEAR READ
1294 ENDCASE
1295 *: EOF: HMCOMP.AC1
1296

```



```

133 IF NOT UVISIBLE("hmet")
134   ACTIVATE WINDOW hmet
135   ENDIF
136 READ CYCLE
137 RELEASE WINDOW hmet
138 #REGION 0
139 IF m.talkstat = "ON"
140   SET TALK ON
141   ENDIF
142 IF m.compstat = "ON"
143   SET COMPATIBLE ON
144   ENDIF
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198

```

HMET/MS-DOS Cleanup Code

```

199 #REGION 1
200 IF m.action = 1
201   SCATTER MEMVAR BLANK
202   m.hmetid=RECCOUNT()+1
203   SHOW GETS
204   SHOW GET action disabled
205   SHOW GET SAVE enabled
206   m.adding = .T.
207 ELSE
208   DO CASE
209     CASE m.action = 2
210       SHOW GETS
211       SHOW GET action disabled
212       SHOW GET SAVE enabled
213     CASE m.action = 3
214       SKIP
215       IF EOF()
216         ?? CHR( 7 )
217         WAIT "Last record" WINDOW NOWAIT
218         SKIP -1
219       ELSE
220         SCATTER MEMVAR
221         SHOW GETS
222       ENDIF
223     CASE m.action = 4
224       SKIP -1
225       IF BOF()
226         ?? CHR( 7 )
227         WAIT "First record" WINDOW NOWAIT
228         SKIP
229       ELSE
230         SCATTER MEMVAR
231         SHOW GETS
232       ENDIF
233     CASE m.action = 5
234       CLEAR READ
235     ENDIF
236   ENDIF
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264

```

HMET/MS-DOS Supporting Procedures and Functions

```

265 #REGION 1
266 PROCEDURE CHANGE
267 *****
268 m.oldexact = SET( "EXACT" )
269 SET EXACT ON
270 m.change =(TRIM(hmet.hmet) <> TRIM( m.hmet))
271 SET EXACT &oldexact
272 RETURN m.change
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298

```

_OKFOVRBIT	m.Action VALID
Function Origin:	
From Platform:	MS-DOS
From Screen:	HMET,
Variable:	m.Action
Called By:	VALID Clause
Object Type:	Push Button
Snippet Number:	1

_OKFOVRBPA	m.Save VALID
Function Origin:	
From Platform:	MS-DOS
From Screen:	HMET,
Variable:	m.Save
Called By:	VALID Clause
Object Type:	Push Button
Snippet Number:	2

```

299 *
300 *
301 *
302 *
303 *
304 *
305 *
306 *
307 *
308 *
309 *
310 *
311 *
312 *
313 *
314 *
315 *
316 *
317 *
318 *
319 *
320 *
321 *
322 *
323 *
324 *
325 *
326 *
327 *
328 *
329 *
330 *
331 *
332 *
333 *
334 *
335 *
336 *
337 *
338 *
339 *
340 *
341 *
342 *
343 *
344 *
345 *
346 *
347 *
348 *
349 *
350 *
351 *
352 *
353 *
354 *
355 *
356 *
357 *
358 *
359 *
360 *
361 *
362 *
363 *
364 *
365 *
366 *
367 *
368 *
369 *
370 *
371 *
372 *
373 *
374 *
375 *
376 *
377 *
378 *
379 *
380 *
381 *
382 *
383 *
384 *
385 *
386 *
387 *
388 *
389 *
390 *
391 *
392 *
393 *
394 *
395 *
396 *
397 *
398 *

```

FUNCTION _okfvrbit	24	m.Action VALID
#REGION 1		
DO CHANGE		

```

265 IF m.save = 1 && Selected Save Button
266
267 IF m.adding && Adding a new record
268     APPEND BLANK
269     GATHER MENVAR
270     ENDIF
271
272 IF m.change && Changing an old record
273     GATHER MENVAR
274     ENDIF
275
276 ELSE
277     SCATTER MENVAR
278     ENDIF
279
280 SHOW GETS
281 SHOW GET action enabled
282 SHOW GET SAVE disabled
283
284 m.adding = .F.
285 m.change = .F.
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304 FUNCTION _qkf0vrbt3    && m.hmet VALID
305 #REGION 1
306 IF m.adding
307     SEEK m.hmet
308     IF FOUND()
309         =errmsg("Record already exists",1)
310         SCATTER MENVAR BLANK FIELD hmet
311     ENDIF
312 ENDIF
313 *: EOF: HMET.AC1

```

_QKFOVRBT3	m.hmet VALID
Function Origin:	
From Platform:	MS-DOS
From Screen:	HMET,
Variable:	m.hmet
Called By:	VALID Clause
Object Type:	Field
Snippet Number:	3
	Record Number: 6

10/27/93	HMLC.SPR	14:48:55
Author's Name Copyright (c) 1993 Company Name Address City, Zip Description: This program was automatically generated by GENSCRN.		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

```

#REGION 0
REGIONAL m.curarea, m.talkstat, m.compstat
IF SET("TALK") = "ON"
  SET TALK OFF
  m.talkstat = "ON"
ELSE
  m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

MS-DOS Window definitions

```

IF NOT WEXIST("hmlc") ;
OR UPPER(UTITLE("HMLC")) = "HMLC.PJX" ;
OR UPPER(UTITLE("HMLC")) = "HMLC.SCX" ;
OR UPPER(UTITLE("HMLC")) = "HMLC.MNX" ;
OR UPPER(UTITLE("HMLC")) = "HMLC.PRG" ;
OR UPPER(UTITLE("HMLC")) = "HMLC.FRX" ;
OR UPPER(UTITLE("HMLC")) = "HMLC.QPR" ;
DEFINE WINDOW hmlc ;
FROM INT((SROW()-12)/2),INT((SCOL()-52)/2) ;
TO INT((SROW()-12)/2)+1,INT((SCOL()-52)/2)+51 ;
NOFLOAT ;
NOCLOSE ;
SHADOW ;
NONMINIMIZE ;
DOUBLE ;
COLOR SCHEME 1
ENDIF

```

HMLC/MS-DOS Setup Code - SECTION 2

```

#REGION 1
PUSH KEY
*ON KEY LABEL ESC DO EscPressed

```

```

67 m.oldscape = SET( "ESCAPE" )
68 SET ESCAPE OFF
69 m.adding = .F.
70 m.change = .F.
71
72 CLOSE ALL
73 SELECT 0
74 USE hmlc
75 SET ORDER TO TAG hmlcid OF hmlc.cdx
76
77 *****
78 * Check see if the last record is defined
79 IF TYPE( "m.LastRec" ) = "N"
80
81 * Start with the first record
82 GO TOP
83 m.lastrec = RECNO()
84
85 ELSE
86 * Start on the last record used
87 GO m.lastrec
88 ENDIF
89 *****
90 SCATTER MEMVAR
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132

```

HMLC/MS-DOS Screen Layout

```

#REGION 1
IF WISIBLE("hmlc")
  ACTIVATE WINDOW hmlc SAME
ELSE
  ACTIVATE WINDOW hmlc NOSHOW
ENDIF
a 1,40 GET m.action ;
  PICTURE "a=VN \<Add;\<Edit;\<Next;\<Previous;\<Exit" ;
  SIZE 1,10,1 ;
  DEFAULT 1 ;
  VALID qkfvvr6jx()
a 1,0 TO 9,39
a 4,11 SAY "a:" ;
  SIZE 1,1,0
a 0,11 SAY "MM LIFE CYCLE PHASES" ;
  SIZE 1,20,0
a 2,3 SAY "PHASE ID:" ;
  SIZE 1,9,0
a 4,1 SAY "PHASE NAME:" ;
  SIZE 1,11,0
a 2,13 GET m.hmlcid ;
  SIZE 1,10 ;
  DEFAULT 0 ;
  DISABLE
a 4,13 GET m.hmlc ;
  SIZE 1,25 ;
  DEFAULT "a" ;
  PICTURE "a:" ;
  VALID qkfvvr6jx() ;
  DISABLE
a 8,10 GET m.save ;
  PICTURE "a=NN \<Save;\<Cancel" ;
  SIZE 1,8,1 ;

```



```

265 * FUNCTION _qkf0vr6te    && m.hmlc VALID
266 #REGION 1
267 IF m.adding
268   m.oldrec = RECNO()
269   GO TOP
270   SEEK m.hmlc
271   IF FOUND()
272     =errmsg("Record already exists",1)
273     SCATTER FIELD hmlc MEMVAR BLANK
274   ENDIF
275   GO m.oldrec
276 ENDIF
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295 * FUNCTION _qkf0vr6vw    && m.Save VALID
296 #REGION 1
297 IF m.save = 1 && Selected Save Button
298   IF m.adding && Adding a new record
299     APPEND BLANK
300     GATHER MEMVAR
301   ELSE
302     DO CHANGE
303     IF m.change && Changing an old record
304       GATHER MEMVAR
305     ENDIF
306   ENDIF
307 ENDIF
308 SCATTER MEMVAR
309 SHOW GETS
310 SHOW GET m.hmlc disabled
311 SHOW GET action enabled
312 SHOW GET SAVE disabled
313
314 m.adding = .F.
315 m.change = .F.
316 *: EOF: HMLC.AC1
317

```

_qkf0vr6vw	m.Save VALID
Function Origin:	
From Platform:	MS-DOS
From Screen:	HMLC,
Variable:	m.Save
Called By:	VALID Clause
Object Type:	Push Button
Snippet Number:	3
	Record Number: 11

10/27/93	HMTAB.SPR	14:49:19
Author's Name		
Copyright (c) 1993 Company Name		
Address		
City, Zip		
Description:		
This program was automatically generated by GENSCRN.		

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

```

#REGION 0
REGIONAL m.curarea, m.talkstat, m.compstat

```

```

IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

MS-DOS Window definitions

```

IF NOT MEXIST("hmtab") ;
OR UPPER(WTITLE("HMTAB")) = "HMTAB.PJX" ;
OR UPPER(WTITLE("HMTAB")) = "HMTAB.SCX" ;
OR UPPER(WTITLE("HMTAB")) = "HMTAB.MNX" ;
OR UPPER(WTITLE("HMTAB")) = "HMTAB.PRG" ;
OR UPPER(WTITLE("HMTAB")) = "HMTAB.FRX" ;
OR UPPER(WTITLE("HMTAB")) = "HMTAB.QPR" ;
DEFINE WINDOW hmtab ;
FROM INT((SROW()-22)/2), INT((SCOL()-79)/2) ;
TO INT((SROW()-22)/2)+21, INT((SCOL()-79)/2)+78 ;
FLOAT ;
NOCLOSE ;
SHADOW ;
nomimize ;
DOUBLE ;
COLOR SCHEME 1
ENDIF

```

HMTAB/MS-DOS Setup Code - SECTION 2

```

#REGION 1
PUSH KEY
*ON KEY LABEL ESC DO EscPressed

```

HMTAB.AC1 12-1-93 11:31a

```

67 m.oldscape = SET("ESCAPE")
68 SET ESCAPE OFF
69 m.adding = .F.
70 m.change = .F.
71 m.action=5
72
73 *****
74 * SELECT HAZARDOUS MATERIAL
75 *****
76 m.data=hmtab()
77 m.hname = dp(m.data,"",1)
78 m.hmatid = INT(VAL(dp(m.data,"",2)))
79 IF EMPTY(m.hname)
80 =ermgs("No material selected",1)
81 RETURN
82 ENDIF
83
84 CLOSE ALL
85 *****
86 * GET UNIT LIST
87 *****
88 IF IUSED("HUNIT")
89 SELECT 0
90 USE hunit ORDER TAG hunit
91 ELSE
92 SELECT hunit
93 ENDIF
94 COPY TO ARRAY unitlist FIELD hunit
95 USE
96 m.unit = "GALLON"
97
98 *****
99 * GET THE MATERIAL RECORDS
100 *****
101 m.hmatid=get_hmid(m.hname)
102 m.hncomid = get_hncomid(m.hmatid)
103 SELECT 0
104 USE hmtab ALIAS hmtab
105 SET FILTER TO hmatid = m.hmatid
106
107 *What if the file is empty?
108 GO TOP
109 m.oldscomid = m.hncomid
110 m.oldsmatid=m.hmatid
111 SCATTER NEWVAR
112 m.oldsunit = m.hunit
113
114 IF m.tabid<1
115 m.action=1
116 m.adding=.T.
117 m.tabid=RECCOUNT()+1
118 m.hmatid=m.oldsmatid
119 m.hncomid = m.oldscomid
120 m.hmlc=""
121 m.oldscomid=0
122 m.oldsmatid=""
123 m.hmet=""
124 m.oldscomid=""
125 m.hmetid=0
126 m.hmetprob=0.000
127 m.hmap=""
128 m.hmapid=0
129
130
131
132

```



```

264 VALID qkf0vrq72() ;
265 DISABLE
266 a 7,18 GET m.hmetid ;
267 SIZE 1,4 ;
268 DEFAULT 0 ;
269 PICTURE "az" ;
270 DISABLE
271 a 7,24 GET m.hmet ;
272 SIZE 1,41 ;
273 DEFAULT " " ;
274 PICTURE "aj" ;
275 WHEN qkf0vrqa1() ;
276 VALID_qkf0vrqc2() ;
277 DISABLE
278 a 8,32 GET m.hmetprob ;
279 SIZE 1,9 ;
280 DEFAULT 0 ;
281 DISABLE
282 a 10,13 GET m.hmcfid ;
283 SIZE 1,3 ;
284 DEFAULT 0 ;
285 PICTURE "az" ;
286 DISABLE
287 a 10,17 GET m.hmcf ;
288 SIZE 1,39 ;
289 DEFAULT " " ;
290 WHEN_qkf0vrqr() ;
291 VALID_qkf0vrqh7() ;
292 DISABLE
293 a 11,4 GET m.element ;
294 PICTURE "a=1HN " ;
295 SIZE 1,9,1 ;
296 DEFAULT 0 ;
297 VALID_qkf0vrqkz() ;
298 DISABLE
299 a 11,13 GET m.hmcfid ;
300 SIZE 1,3 ;
301 DEFAULT 0 ;
302 PICTURE "az" ;
303 DISABLE
304 a 11,17 GET m.hmcf ;
305 SIZE 1,39 ;
306 DEFAULT " " ;
307 PICTURE "aj" ;
308 DISABLE
309 a 12,13 GET m.hmcfiid ;
310 SIZE 1,3 ;
311 DEFAULT 0 ;
312 PICTURE "az" ;
313 DISABLE
314 a 12,17 GET m.hmcfid ;
315 SIZE 1,39 ;
316 DEFAULT " " ;
317 PICTURE "aj" ;
318 DISABLE
319 a 14,13 GET m.hmcfecost ;
320 SIZE 1,9 ;
321 DEFAULT 0 ;
322 PICTURE "$$$,$$$,$$$,$$$$," ;
323 DISABLE
324 a 14,46 GET m.prob ;
325 SIZE 1,5 ;
326 DEFAULT 0 ;
327 PICTURE "no.99" ;
328 DISABLE
329 a 16,33 GET m.pdp ;

```

```

330 PICTURE "a*RNH YES:NO";
331 SIZE 1,7,0;
332 DEFAULT 2;
333 DISABLE
334 @ 17,33 GET m.perd;
335 PICTURE "a*RNH YES:NO";
336 SIZE 1,7,0;
337 DEFAULT 2;
338 DISABLE
339 @ 18,33 GET m.perq;
340 PICTURE "a*RNH YES:NO";
341 SIZE 1,7,0;
342 DEFAULT 2;
343 DISABLE
344 @ 16,68 GET m.save;
345 PICTURE "a*VN \Save;\<Cancel";
346 SIZE 1,8,1;
347 DEFAULT 1;
348 VALID okf0vrqrg();
349 DISABLE
350 @ 11,4 SAY "Element:";
351 SIZE 1,8,0
352 @ 9,3 TO 19,63
353 @ 14,33 SAY "Probability:";
354 SIZE 1,11,0
355 @ 16,19 SAY "By Person";
356 SIZE 1,9,0
357 @ 1,4 SAY "Material:";
358 SIZE 1,9,0
359 @ 0,26 SAY "HAZARDOUS MATERIALS TABLE";
360 SIZE 1,25,0
361 @ 4,4 SAY "Phase:";
362 SIZE 1,6,0
363 @ 5,4 SAY "Process:";
364 SIZE 1,8,0
365 @ 7,4 SAY "Exposure Type:";
366 SIZE 1,14,0
367 @ 8,4 SAY "Probability of exposure:";
368 SIZE 1,28,0
369 @ 12,7 SAY "Item:";
370 SIZE 1,5,0
371 @ 14,7 SAY "Cost:";
372 SIZE 1,5,0
373 @ 10,5 SAY "Factor:";
374 SIZE 1,7,0
375 @ 0,4 SAY "H";
376 SIZE 1,1,0
377 @ 2,42 SAY "Per:";
378 SIZE 1,4,0
379 @ 18,19 SAY "By Quantity";
380 SIZE 1,11,0
381 @ 17,19 SAY "By Day";
382 SIZE 1,6,0
383
384 [ IF NOT WVISIBLE("hmtab")
385   ACTIVATE WINDOW hmtab
386 ]
387 ENDIF
388 READ CYCLE
389
390 RELEASE WINDOW hmtab
391
392 #REGION 0
393 [ IF M.talkstat = "QM"
394   SET TALK ON
395 ]
396 ENDIF

```

```

396 IF m.compstat = "OK"
397   SET COMPATIBLE ON
398 ENDIF

```

```

400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461

```

HMTAB/MS-DOS Cleanup Code

```

407 #REGION 1
408 CLOSE ALL
409 POP KEY ALL
410 SET ESCAPE &oldescape
411 RETURN
412 ***** End of Main Body - Entry Cleanup
413 *****
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461

```

HMTAB/MS-DOS Supporting Procedures and Functions

```

425 #REGION 1
426 PROCEDURE espresced
427 RETURN
428 *****
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461

```

```

430 *****
431 PROCEDURE CHANGE
432 *****
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461

```

```

433 IF USED("HMTAB")
434   SELECT hmtab
435   SET FILTER TO hmatid = m.hmatid
436 ELSE
437   SELECT 0
438   USE hmtab ALIAS hmtab
439   SET FILTER TO hmatid = m.hmatid
440 ENDIF
441
442 m.oldexact = SET( "EXACT" )
443 SET EXACT ON
444 m.change = (hmtab.hmatid <> m.hmatid;
445 OR hmtab.hmtcid <> m.hmtcid;
446 OR hmtab.hmcpid <> m.hmcpid;
447 OR hmtab.hmetid <> m.hmetid;
448 OR hmtab.hmetprob <> m.hmetprob;
449 OR hmtab.hmcfid <> m.hmcfid;
450 OR hmtab.hmcfid <> m.hmcfid;
451 OR hmtab.hmcfid <> m.hmcfid;
452 OR hmtab.hmcfecost <> m.hmcfecost;
453 OR hmtab.hmunit <> m.hmunit;
454 OR hmtab.prob <> m.prob;
455 OR hmtab.waverage <> m.waverage;
456 OR hmtab.perd <> m.perd;
457 OR hmtab.perq <> m.perq;
458 OR hmtab.perp <> m.perp)
459 SET EXACT &oldexact
460 RETURN m.change
461

```

```

462 *****
463 *FUNCTION GET_HNAME
464 *****
465 *RELEASE hmn
466 *DIMENSION hmn(1)
467 *hmn(1)=""
468 *M.oldfile=SELECT()
469
470 *SELECT hmat.hmatname;
471 *   FROM hmat;
472 *   WHERE hmat.hmatname IN (alltrim(m.hname));
473 *   INTO ARRAY hmn
474
475 *m.ans=""
476 *IF NOT EMPTY(hmn(1))
477 *   m.ans=CHOOSE(2,hmn,"Select a Material")
478 *ELSE
479 *   =ERRMSG(m.hname + " was not found",1)
480 *   m.ans=""
481 *ENDIF
482
483 *SELECT (M.oldfile)
484 *RETURN m.ans
485
486 *****
487 *FUNCTION get_hmid
488 *****
489 *PARAMETER m.hname
490
491 *m.oldfile=SELECT()
492
493 *SELECT DISTINCT hmat.hmatid;
494 *   FROM hmat;
495 *   WHERE hmat.hmatname IN (alltrim(m.hname));
496 *   INTO ARRAY X
497
498 *M.HMATID=X(1)
499 *SELECT (m.oldfile)
500 *RETURN M.HMATID
501
502 *****
503 *FUNCTION get_hmcomid
504 *****
505 *PARAMETER m.hmatid
506
507 m.oldfile=SELECT()
508
509 SELECT DISTINCT hmat.niin;
510 FROM hmat;
511 WHERE hmat.hmatid = m.hmatid;
512 INTO ARRAY X
513 IF TYPE("X") != "N"
514   m.stock = X(1)
515 ELSE
516   m.stock = 0
517 ENDIF
518 IF IUSED("HMCOM")
519   SELECT 0
520   USE hmcom
521   SELECT hmcom
522 ELSE
523   SELECT hmcom
524 ENDIF
525 SET ORDER TO TAG niin
526 SEEK m.stock
527 IF FOUND()

```

```

528 m.comid = hmc.comid
529 ELSE
530 m.comid = 0
531 ENDIF
532 SELECT (m.oldfile)
533 RETURN m.comid
534
535 *****
536 *GET LIFE CYCLE PHASE
537 FUNCTION get_hmlc
538 *****
539 PARAMETER m.hmlc
540
541 RELEASE hmn
542 DIMENSION hmn[1]
543 hmn[1]=""
544 m.oldfile=SELECT()
545
546 SELECT hmlc.hmlc;
547 FROM hmlc;
548 WHERE hmlc.hmlc IN (ALLTRIM(m.hmlc));
549 ORDER BY hmlc;
550 INTO ARRAY hmn
551
552 m.ans=""
553 IF NOT EMPTY(hmn[1])
554 m.ans=chooser(hmn,"Select a Life Cycle Phase")
555 ELSE
556 =errmsg(m.hmlc + " was not found",1)
557 m.ans=""
558 ENDIF
559
560 SELECT (m.oldfile)
561 RETURN m.ans
562
563 *****
564 FUNCTION get_hmlcn
565 PARAMETER m.hmlcid
566 m.oldfile=SELECT()
567
568 SELECT DISTINCT hmlc.hmlc;
569 FROM hmlc;
570 WHERE hmlc.hmlcid = (m.hmlcid);
571 INTO ARRAY X
572
573 m.hmlc=X[1]
574 SELECT (m.oldfile)
575 RETURN m.hmlc
576
577 *****
578 *Working Processes
579 FUNCTION get_hmp
580 *****
581 PARAMETER m.hmp
582 RELEASE hmn
583 DIMENSION hmn[1]
584 hmn[1]=""
585 m.oldfile=SELECT()
586
587 SELECT hmp.hmp;
588 FROM hmp;
589 WHERE hmp.hmp IN (ALLTRIM(m.hmp));
590 INTO ARRAY hmn
591
592 m.ans=""
593 IF NOT EMPTY(hmn[1])

```

```

594 m.ans=chooser(hmn,"Select a Process")
595 ELSE
596 =errmsg(m.hmp + " was not found",1)
597 m.ans=""
598 ENDIF
599
600 SELECT (m.oldfile)
601 RETURN m.ans
602
603 *****
604 FUNCTION get_hmapn
605 PARAMETER m.hmapid
606 m.oldfile=SELECT()
607
608 SELECT DISTINCT hmap.hmap;
609 FROM hmap;
610 WHERE hmap.hmapid = m.hmapid;
611 INTO ARRAY X
612
613 m.hmap=X[1]
614 SELECT(m.oldfile)
615 RETURN m.hmap
616
617 *****
618 FUNCTION get_hmet
619 *****
620 PARAMETER m.hmet
621
622 RELEASE hmn
623 DIMENSION hmn[1]
624 hmn[1]=""
625 m.oldfile=SELECT()
626
627 SELECT hmet.hmet;
628 FROM hmet;
629 WHERE hmet.hmet IN (ALLTRIM(m.hmet));
630 INTO ARRAY hmn
631
632 m.ans=""
633 IF NOT EMPTY(hmn[1])
634 m.ans=chooser(hmn,"Select an Exposure Type")
635 ELSE
636 =errmsg(m.hmet + " was not found",1)
637 ENDIF
638
639 SELECT (m.oldfile)
640 RETURN m.ans
641
642 *****
643 FUNCTION get_hmetid
644 *****
645 PARAMETER m.hmet
646 m.oldfile=SELECT()
647
648 SELECT DISTINCT hmet.hmetid;
649 FROM hmet;
650 WHERE hmet.hmet IN (ALLTRIM(m.hmet));
651 INTO ARRAY X
652
653 m.hmetid=X[1]
654 SELECT (m.oldfile)
655 RETURN m.hmetid
656
657 *****
658 FUNCTION get_hmetn

```

```

660 *****
661 PARAMETER m.hmcid
662 m.oldfile=SELECT()
663
664 SELECT DISTINCT hmet.hmet;
665 FROM hmet;
666 WHERE hmet.hmetid = (m.hmetid);
667 INTO ARRAY X
668
669 m.hmet=X[1]
670 SELECT (m.oldfile)
671 RETURN m.hmet
672
673 ***** COST FACTORS *****
674
675 *****
676 FUNCTION get_cfar
677 *get the cost factors
678 *****
679 PARAMETER m.hmcid
680 m.oldfile=SELECT()
681
682 DECLARE cfarr[1]
683 cfarr[1]=" "
684
685 SELECT hmcid.hmcid;
686 FROM hmcid;
687 WHERE hmcid.hmcid = (m.hmcid);
688 INTO ARRAY cfarr
689
690 m.hmcid=cfarr[1]
691 SELECT(m.oldfile)
692 RETURN m.hmcid
693
694 *****
695 FUNCTION get_cfar1
696 *get cost factor elements
697 *****
698 PARAMETER m.hmcfeid
699 m.oldfile=SELECT()
700 DECLARE cfarr1[1]
701 cfarr1[1]=" "
702
703 SELECT hmcfe.hmcfe;
704 FROM hmcfe;
705 WHERE hmcfe.hmcfeid = (m.hmcfeid);
706 INTO ARRAY cfarr1
707
708 m.hmcfe=cfarr1[1]
709 SELECT (m.oldfile)
710 RETURN m.hmcfe
711
712 *****
713 FUNCTION get_cf
714 *****
715 PARAMETER m.hmcfe
716
717 RELEASE MEMO LIKE cfarr
718 DIMENSION cfarr[1]
719 cfarr[1]=" "
720
721 m.oldfile=SELECT()
722 SELECT hmcid.hmcid;
723 FROM hmcid;
724
725
726 WHERE hmcid.hmcid IN (ALLTRIM(m.hmcid));
727 ORDER BY hmcid.hmcid;
728 INTO ARRAY cfearr
729
730 IF NOT EMPTY(cfearr[1])
731 m.x=chooser(cfearr,"Select a Cost Factor")
732 ELSE
733 zerrmsg(m.x + " was not found",1)
734 m.hmcfeid=0
735 m.x=""
736
737 ENDIF
738
739 SELECT (m.oldfile)
740 RETURN m.x
741
742 *****
743 FUNCTION get_hmcid
744 *****
745 PARAMETER m.hmcid
746 m.oldfile = SELECT()
747 SELECT DISTINCT hmcid.hmcid;
748 FROM hmcid;
749 WHERE hmcid.hmcid IN (ALLTRIM(m.hmcid));
750 INTO ARRAY X
751 m.hmcid=X[1]
752 SELECT (m.oldfile)
753 RETURN m.hmcid
754
755 *****
756 FUNCTION get_cfi
757 *****
758 PARAMETER m.hmcfe
759
760 RELEASE MEMO LIKE cfearr1
761 DIMENSION cfearr1[1]
762 cfearr1[1]=" "
763 m.oldfile=SELECT()
764
765 SELECT hmcfe.hmcfe;
766 FROM hmcfe;
767 WHERE hmcfe.hmcfe IN (ALLTRIM(m.hmcfe));
768 ORDER BY hmcfe.hmcfeid;
769 INTO ARRAY cfearr1
770
771 IF NOT EMPTY(cfearr1[1])
772 m.x=chooser(cfearr1,"Select a Cost Factor Element")
773 ELSE
774 zerrmsg(m.x + " was not found",1)
775 m.hmcfeid=0
776 m.x=""
777
778 ENDIF
779 SELECT (m.oldfile)
780 RETURN m.x
781
782 *****
783 FUNCTION get_eid
784 *****
785 PARAMETER m.hmcfe
786
787 m.oldfile = SELECT()
788
789
790
791

```

```

792 SELECT DISTINCT hmcfe.hmcfeid;
793 FROM hmcfe;
794 WHERE hmcfe.hmcfe IN (ALLTRIM(m.hmcfe));
795 INTO ARRAY X
796
797 m.hmcfeid=X[1]
798 SELECT (m.oldfile)
799 RETURN m.hmcfeid
800
801 *****
802 FUNCTION get_ei
803 *****
804 PARAMETER m.hmcfeid
805 m.oldfile=SELECT()
806
807 SELECT hmcfe.hmcfeid;
808 FROM hmcfe;
809 WHERE hmcfe.hmcfeid = (m.hmcfeid);
810 INTO ARRAY X
811
812 m.hmcfeid=X[1]
813 SELECT (m.oldfile)
814 RETURN m.hmcfeid
815
816 *****
817 FUNCTION get_eid
818 *****
819 PARAMETER m.hmcfeid
820 m.oldfile=SELECT()
821
822 SELECT hmcfe.hmcfeid;
823 FROM hmcfe;
824 WHERE hmcfe.hmcfeid IN (ALLTRIM(m.hmcfeid));
825 INTO ARRAY X
826
827 m.hmcfeid=X[1]
828 SELECT (m.oldfile)
829 RETURN m.hmcfeid
830
831 *****
832 FUNCTION rel
833 *****
834 PARAMETER m.id
835 EXTERNAL ARRAY cost
836 m.oldfile=SELECT()
837
838 IF PARAMETER()=0
839 m.id=0
840 ENDIF
841
842 SET TALK OFF
843
844 &OPEN FILE # 1
845
846 IF USED("CFEITMP")
847 SELECT cfeitmp
848 SET ORDER TO TAG hmcfeid OF hmcfe.cdx
849
850 ELSE
851 SELECT 0
852 USE hmcfe ALIAS cfeitmp AGAIN
853 SET ORDER TO TAG hmcfeid OF hmcfe.cdx
854
855 ENDIF
856
857

```

```

858 &OPEN FILE # 2
859
860 IF USED("CFEITMP")
861 SELECT cfeitmp
862 SET ORDER TO TAG hmcfeid OF hmcfe.cdx
863
864 ELSE
865 SELECT 0
866 USE hmcfe ALIAS cfeitmp AGAIN
867 SET ORDER TO TAG hmcfeid OF hmcfe.cdx
868
869 ENDIF
870
871 &OPEN FILE #3
872
873 IF USED("CFTMP")
874 SELECT cftmp
875 SET ORDER TO TAG hmcfeid OF hmcfe.cdx
876
877 ELSE
878 SELECT 0
879 USE hmcfe ALIAS cftmp AGAIN
880 SET ORDER TO TAG hmcfeid OF hmcfe.cdx
881
882 ENDIF
883
884 IF m.id>0
885 SET FILTER TO hmcfeid = m.id
886
887 ENDIF
888
889 SELECT cftmp
890 SET RELATION TO hmcfeid INTO cfeitmp ADDITIVE
891
892 SELECT cftmp
893 SET RELATION TO hmcfeid INTO cftmp ADDITIVE
894
895 SET SKIP TO cfeitmp, cftmp
896
897 ** Show fields from grandparent (HMCFEI), parent (HMCFE) and child (H
898 => MCF)**
899 BROWSE FIELDS cftmp.hmcfe:H="FACTORS", cftmp.hmcfe:H="ELEMENTS";
900 cfeitmp.hmcfeid:H="ITEMS", cfeitmp.hmcfeid:H="COSTS", cfeitmp.h
901 => mcfunit:H="UNITS";
902 NOMODIFY NOAPPEND MODELETE NORMAL TITLE "COST FACTORS"
903
904 SCATTER FIELDS cftmp.hmcfe, cftmp.hmcfe, cfeitmp.hmcfe, cfeitmp.hmcfe, cfeitmp.hmcfe
905 => eicost, cfeitmp.hmcfeunit;
906 TO cost
907
908 SET RELATION TO
909 IF USED('cftmp')
910 SELECT cftmp
911 USE
912 ENDIF
913 IF USED('cfeitmp')
914 SELECT cfeitmp
915 USE
916 ENDIF
917 IF USED('cfeitmp')
918 SELECT cfeitmp
919 USE
920 ENDIF
921 *CLOSE DATABASE CFTMP,CFTMP,CFTMP,CFTMP
922 SELECT (m.oldfile)

```


921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986

```
-CASE m.action = 2  
    SHOW GET  
    SHOW GET action  
    SHOW GET m.elem  
    SHOW GET SAVE e  
    SHOW GET m.hmcid  
    SHOW GET m.hmcpsd  
    SHOW GET m.hwst  
    SHOW GET m.hwstt  
    SHOW GET m.hwcf  
    SHOW GET m.hwcfi  
    SHOW GET m.prob  
    SHOW GET m.wtsave  
    * SHOW GET  
    SHOW GET m.perfip  
    SHOW GET m.percd  
    SHOW GET m.percd
```

```

--CASE m.action = 3
  -IF EOF()
    SKIP
  -ENDIF
  -IF EOF()
    ?? CHR( 7 )
    WAIT "Last record" WINDOW NOWAIT
    GO BOTTOM
  -ENDIF

--CASE m.action = 4
  -IF EOF()
    SKIP -1
  -ENDIF
  -IF EOF()
    ?? CHR( 7 )
    WAIT "First record" WINDOW NOWAIT
    GO TOP
  -ENDIF

--CASE m.action = 5
  GO TOP

--CASE m.action=6
  GO BOTTOM

--CASE m.action = 7
  CLEAR READ
  -ENDCASE
DIF

m.action > 2 AND m.action < 7
  SCATTER MENVAR

-IF m.hmlcid>0
  m.hmlc=get_hmlcn(m.hmlcid)
  m.otdhlmc=m.hmlc
-ELSE
  m.hmlc=""
  m.otdhlmc=""
-ENDIF

-IF m.hmlcpid>0
  m.hmlcpset_hmlcpn(m.hmlcpid)

```

```
-if m.hmap[id>0
m.hmap[id]=get
hmapson(m.hmap[id)
```

```

1053 m.oldhmap=m.hmap
1054 ELSE
1055 m.hmap=""
1056 m.oldhmap=""
1057 ENDIF
1058
1059 IF m.hmetid>0
1060 m.hmet=get_hmetn(m.hmetid)
1061 ELSE
1062 m.hmet=""
1063 m.oldhmet=""
1064 ENDIF
1065
1066 IF m.hmcfid>0
1067 m.hmcf=get_cfar(m.hmcfid)
1068 m.oldhmcf=m.hmcf
1069 ELSE
1070 m.hmcf=""
1071 m.oldhmcf=""
1072 ENDIF
1073
1074 IF m.hmcfeid>0
1075 m.hmcf=get_cfar1(m.hmcfeid)
1076 ELSE
1077 m.hmcf=""
1078 ENDIF
1079
1080 IF m.hmcfeid>0
1081 m.hmcf=get_ei(m.hmcfeid)
1082 ELSE
1083 m.hmcf=""
1084 ENDIF
1085
1086 SHOW GETS
1087 SHOW GET m.element disabled
1088 SHOW GET m.save disabled
1089 SHOW GET m.hmc disabled
1090 SHOW GET m.hmap disabled
1091 SHOW GET m.hmet disabled
1092 SHOW GET m.hmcf disabled
1093 SHOW GET m.hmcfecost disabled
1094 SHOW GET m.mprob disabled
1095 SHOW GET m.mhunit DISABLED
1096 SHOW GET m.perp disabled
1097 SHOW GET m.perq disabled
1098 ENDIF

```

_QKFOVRPYS
 Function Origin:
 From Platform: MS-DOS
 From Screen: HMTAB,
 Variable: m.hmc
 Called By: WHEN Clause
 Object Type: Field
 Record Number: 8

HMTAB.AC1 12-1-93 11:31a

```

1119 *
1120 *
1121 *
1122 FUNCTION _qkfvvrpys 22 m.hmc WHEN
1123 #REGION 1
1124 m.oldhmc=m.hmc
1125
1126 *
1127 *
1128 *
1129 *
1130 *
1131 *
1132 *
1133 *
1134 *
1135 *
1136 *
1137 *
1138 *
1139 *
1140 *
1141 *
1142 *
1143 *
1144 *
1145 *
1146 *
1147 *
1148 *
1149 *
1150 *
1151 *
1152 *
1153 *
1154 *
1155 *
1156 *
1157 *
1158 *
1159 *
1160 *
1161 *
1162 *
1163 *
1164 *
1165 *
1166 *
1167 *
1168 *
1169 *
1170 *
1171 *
1172 *
1173 *
1174 *
1175 *
1176 *
1177 *
1178 *
1179 *
1180 *
1181 *
1182 *
1183 *
1184 *

```

_QKFOVRQ1C
 Function Origin:
 From Platform: MS-DOS
 From Screen: HMTAB,
 Variable: m.hmc
 Called By: VALID Clause
 Object Type: Field
 Snippet Number: 3
 Record Number: 8

```

FUNCTION _qkfvvrq1c 22 m.hmc VALID
#REGION 1
IF m.oldhmc <> m.hmc
  IF NOT EMPTY(m.hmc)
    m.hmc=ALLTRIM(m.hmc)
    m.hmc=IFF(m.hmc="?", "", UPPER(m.hmc))
    m.hmc=get_hmc(m.hmc)
    IF NOT EMPTY(m.hmc)
      SELECT DISTINCT hmc.hmcid;
      FROM hmc;
      WHERE hmc.hmc IN (m.hmc);
      INTO ARRAY X
      m.hmcid=X[1]
    ENDIF
  ENDIF
ENDIF
SHOW GETS
ENDIF

```

_QKFOVRQ4T
 Function Origin:
 From Platform: MS-DOS
 From Screen: HMTAB,
 Variable: m.hmap
 Called By: WHEN Clause
 Object Type: Field
 Snippet Number: 4
 Record Number: 10

```

FUNCTION _qkfvvrq4t 22 m.hmap WHEN
#REGION 1
m.oldhmap=m.hmap

```

Page 9 of 12

```

1185 *
1186 *
1187 *
1188 *
1189 *
1190 *
1191 *
1192 *
1193 *
1194 *
1195 *
1196 *
1197 *
1198 *
1199 *
1200 *
1201 *
1202 *
1203 *
1204 *
1205 *
1206 *
1207 *
1208 *
1209 *
1210 *
1211 *
1212 *
1213 *
1214 *
1215 *
1216 *
1217 *
1218 *
1219 *
1220 *
1221 *
1222 *
1223 *
1224 *
1225 *
1226 *
1227 *
1228 *
1229 *
1230 *
1231 *
1232 *
1233 *
1234 *
1235 *
1236 *
1237 *
1238 *
1239 *
1240 *
1241 *
1242 *
1243 *
1244 *
1245 *
1246 *
1247 *
1248 *
1249 *
1250 *

```

_OKFOVRQ2 m.hmap VALID

Function Origin:

From Platform: MS-DOS Record Number: 10

From Screen: INITAB,

Variable: m.hmap

Called By: VALID Clause

Object Type: Field

Snippet Number: 5

```

1201 IF m.olchmap <> m.hmap
1202   m.hmap=ALLTRIM(m.hmap)
1203   m.hmap=IF(m.hmap="*", "", UPPER(m.hmap))
1204   m.hmap=ALLTRIM(get_hmap(m.hmap))
1205   IF NOT EMPTY(m.hmap)
1206     SELECT hmap.hmapid;
1207     FROM hmap;
1208     WHERE hmap.hmap IN (m.hmap);
1209     INTO ARRAY X
1210     m.hmapid=X[1]
1211   ELSE
1212     m.hmap = SPACE(80)
1213   ENDIF
1214 SHOW GETS
1215 ENDIF

```

_OKFOVRQAL m.hmet WHEN

Function Origin:

From Platform: MS-DOS Record Number: 12

From Screen: INITAB,

Variable: m.hmet

Called By: WHEN Clause

Object Type: Field

Snippet Number: 6

```

1234 FUNCTION _okfovraqal      && m.hmet WHEN
1235 #REGION 1
1236 m.olchmet=m.hmet

```

_OKFOVRQ2 m.hmet VALID

Function Origin:

From Platform: MS-DOS Record Number: 12

From Screen: INITAB,

Variable: m.hmet

Called By: VALID Clause

Object Type: Field

Snippet Number: 7

```

1251 *
1252 *
1253 *
1254 *
1255 *
1256 *
1257 *
1258 *
1259 *
1260 *
1261 *
1262 *
1263 *
1264 *
1265 *
1266 *
1267 *
1268 *
1269 *
1270 *
1271 *
1272 *
1273 *
1274 *
1275 *
1276 *
1277 *
1278 *
1279 *
1280 *
1281 *
1282 *
1283 *
1284 *
1285 *
1286 *
1287 *
1288 *
1289 *
1290 *
1291 *
1292 *
1293 *
1294 *
1295 *
1296 *
1297 *
1298 *
1299 *
1300 *
1301 *
1302 *
1303 *
1304 *
1305 *
1306 *
1307 *
1308 *
1309 *
1310 *
1311 *
1312 *
1313 *
1314 *
1315 *
1316 *

```

_OKFOVRQ2 m.hmet VALID

Function Origin:

From Platform: MS-DOS Record Number: 15

From Screen: INITAB,

Variable: m.hmet

Called By: WHEN Clause

Object Type: Field

Snippet Number: 8

```

1251 FUNCTION _okfovraq2      && m.hmet VALID
1252 #REGION 1
1253 IF m.olchmet <> m.hmet
1254   m.hmet=ALLTRIM(m.hmet)
1255   m.hmet=IF(m.hmet="*", "", UPPER(m.hmet))
1256   m.hmet=ALLTRIM(get_hmet(m.hmet))
1257   IF NOT EMPTY(m.hmet)
1258     SELECT DISTINCT hmet.hmetid;
1259     FROM hmet;
1260     WHERE hmet.hmet IN (m.hmet);
1261     INTO ARRAY X
1262     m.hmetid=X[1]
1263   ENDIF
1264 SHOW GETS
1265 ENDIF

```

_OKFOVRQH7 m.hmacf VALID

Function Origin:

From Platform: MS-DOS Record Number: 15

From Screen: INITAB,

Variable: m.hmacf

Called By: VALID Clause

Object Type: Field

Snippet Number: 9

```

1290 FUNCTION _okfovraqh7      && m.hmacf WHEN
1291 #REGION 1
1292 m.olchmacf=m.hmacf

```

_OKFOVRQH7 m.hmacf VALID

Function Origin:

From Platform: MS-DOS Record Number: 15

From Screen: INITAB,

Variable: m.hmacf

Called By: VALID Clause

Object Type: Field

Snippet Number: 9

```

1310 FUNCTION _okfovraqh7      && m.hmacf VALID
1311 #REGION 1
1312 IF m.olchmacf <> m.hmacf
1313   m.hmacf=ALLTRIM(m.hmacf)
1314   m.hmacf=IF(m.hmacf="*", "", UPPER(m.hmacf))
1315   m.hmacf=ALLTRIM(get_cf(m.hmacf))

```



```
1449      m.hmcfejs=""
1450  END IF
1451  SHOW GETS
1452  SHOW GET m.action enabled
1453  SHOW GET m.element disabled
1454  SHOW GET SAVE disabled
1455  SHOW GET m.hmlc disabled
1456  SHOW GET m.hmap disabled
1457  SHOW GET m.hmet disabled
1458  SHOW GET m.hmetprob disabled
1459  SHOW GET m.hmcf disabled
1460  SHOW GET m.hmcfecost disabled
1461  SHOW GET m.prob disabled
1462  SHOW GET m.hmunit disabled
1463  SHOW GET m.perp disabled
1464  SHOW GET m.perd disabled
1465  SHOW GET m.perq disabled
1466  m.adding=.f.
1467  m.charge=.f.
1468  *: EOF: HMTAB.AC1
```

10/27/93	HMWP.SPR	14:48:59
Author's Name Copyright (c) 1993 Company Name Address City, Zip Description: This program was automatically generated by GENSCRN.		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *

```

```

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

```

```

IF SET("TALK") = "ON"
  SET TALK OFF
  m.talkstat = "ON"
ELSE
  m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

MS-DOS window definitions

```

38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

```

IF NOT WEXIST("hmwp");
  OR UPPER(WTITLE("HMWP")) = "HMWP.PJX";
  OR UPPER(WTITLE("HMWP")) = "HMWP.SCX";
  OR UPPER(WTITLE("HMWP")) = "HMWP.MNX";
  OR UPPER(WTITLE("HMWP")) = "HMWP.PRG";
  OR UPPER(WTITLE("HMWP")) = "HMWP.FRX";
  OR UPPER(WTITLE("HMWP")) = "HMWP.QPR";
  DEFINE WINDOW hmwp;
  FROM INT((SROW()-15)/2), INT((SCOL()-64)/2);
  TO INT((SROW()-15)/2)+14, INT((SCOL()-64)/2)+63;
  NOFLOAT;
  NOCLOSE;
  SHADOW;
  minimize;
  DOUBLE;
  COLOR SCHEME 1
ENDIF

```

HMWP/MS-DOS Setup Code - SECTION 2

```

#REGION 1
PUSH KEY
*ON KEY LABEL ESC DO EscPressed

```

```

67 m.oldscape = SET( "ESCAPE" )
68 SET ESCAPE OFF
69 m.adding = .F.
70 m.change = .F.
71
72 IF IUSED("HMWP")
73   SELECT 0
74   USE hmwp
75   SET ORDER TO TAG hmwp OF hmwp.cdx
76 ELSE
77   SELECT hmwp
78   SET ORDER TO TAG hmwp
79 ENDIF
80
81 *****
82 * Check see if the last record is defined
83 IF TYPE( "m.LastRec" ) = "U"
84
85   * Start with the first record
86   GO TOP
87   m.lastrec = RECNO()
88
89 ELSE
90   * Start on the last record used
91   GO m.lastrec
92 ENDIF
93 *****
94 SCATTER MENVAR
95
96 *
97 *
98 *
99 *
100 *
101 *
102 *
103 *
104 *
105 #REGION 1
106 IF WVISIBL("hmwp")
107   ACTIVATE WINDOW hmwp SAME
108 ELSE
109   ACTIVATE WINDOW hmwp NOSHOW
110 ENDIF
111 @ 2,50 GET m.action;
112   PICTURE "g*vn \<Add;\<Edit;\<Next;\<Previous;\<7E\<xit" ;
113   SIZE 1,10,1;
114   DEFAULT 1;
115   VALID qkf0vr92i();
116   @ 2,11 GET m.hmwpid;
117   SIZE 1,10;
118   DEFAULT " ";
119   DISABLE
120   @ 4,11 GET m.hmwp;
121   SIZE 6,34;
122   DEFAULT " ";
123   PICTURE "g1";
124   VALID qkf0vr99i();
125   DISABLE
126   @ 11,14 GET m.save;
127   PICTURE "g*nn \<Save;\<Cancel" ;
128   SIZE 1,8,1;
129   DEFAULT 1;
130   VALID qkf0vr9c5();
131   DISABLE
132   @ 1,0 TO 12,46
133   @ 2,3 SAY "ID NUM:";

```

HMWP/MS-DOS Screen Layout

```

133 SIZE 1,7,0
134 @ 4,2 SAY "PROCESS:" ;
135 SIZE 1,8,0
136 @ 0,16 SAY "HM PROCESSES" ;
137 SIZE 1,12,0

```

```

138 IF NOT WISIBLE("hmwp")
139 ACTIVATE WINDOW hmwp
140 ENDIF

```

```

141 READ CYCLE

```

```

142 RELEASE WINDOW hmwp

```

```

143 #REGION 0

```

```

144 IF m.talkstat = "ON"
145 SET TALK ON
146 ENDIF

```

```

147 IF m.compstat = "ON"
148 SET COMPATIBLE ON
149 ENDIF

```

HMWP/MS-DOS Cleanup Code

```

150 #REGION 1
151 IF USED('HMWP')
152 USE
153 ENDIF

```

```

154 POP KEY ALL
155 SET ESCAPE &oldescape
156 ***** End of Main Body - Entry Cleanup
157 *****

```

HMWP/MS-DOS Supporting Procedures and Functions

```

158 #REGION 1
159 PROCEDURE CHANGE
160 *****
161 m.oldexact = SET( "EXACT" )
162 SET EXACT ON
163 m.change = TRIM(hmwp.hmwp) <> TRIM( m.hmwp )
164 SET EXACT &oldexact
165 RETURN m.change

```

```

166 _OKFOVR921 m.Action VALID

```

```

167 Function Origin:

```

```

168 From Platform: MS-DOS
169 From Screen: HMWP, Record Number: 2

```

```

199 * Variable: m.Action
200 * Called By: VALID Clause
201 * Object Type: Push Button
202 * Snippet Number: 1

```

```

203 FUNCTION _okf0vr92i m.m.Action VALID

```

```

204 #REGION 1
205 IF m.action = 1
206 SCATTER MEMVAR BLANK
207 m.hmwpid=RECCOUNT()-1
208 SHOW GETS
209 SHOW GET m.hmwp enabled
210 SHOW GET action disabled
211 SHOW GET SAVE enabled
212 m.adding = .T.

```

```

213 ELSE

```

```

214 DO CASE

```

```

215 CASE m.action = 2
216 SHOW GETS
217 SHOW GET m.hmwp enabled
218 SHOW GET action disabled
219 SHOW GET SAVE enabled

```

```

220 CASE m.action = 3

```

```

221 SKIP
222 IF EOF()
223 ?? CHR( 7 )
224 WAIT "Last record" WINDOW NOWAIT
225 SKIP -1
226 ELSE
227 SCATTER MEMVAR
228 SHOW GETS
229 ENDIF

```

```

230 CASE m.action = 4

```

```

231 IF BOF()
232 ?? CHR( 7 )
233 WAIT "First record" WINDOW NOWAIT
234 SKIP
235 ELSE
236 SCATTER MEMVAR
237 SHOW GETS
238 ENDIF

```

```

239 CASE m.action = 5

```

```

240 CLEAR READ
241 ENDCASE
242 ENDIF

```

```

243 _OKFOVR991 m.hmwp VALID

```

```

244 Function Origin:

```

```

245 From Platform: MS-DOS
246 From Screen: HMWP, Record Number: 4
247 Variable: m.hmwp
248 Called By: VALID Clause

```

```

331
332
333
334
335
336

```

SHOW GET SAVE disabled
m.adding = .f.
m.change = .f.
ENDIF
*: EOF: HMAP.AC1

```

265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330

```

Object Type: Field
Snippet Number: 2

FUNCTION _qkf0vr99i && m.hmap VALID
#REGION 1
IF m.adding
m.oldrec = RECNO()
SEEK m.hmap
IF FOUND()
=errmsg("Record already exists",1)
SCATTER FIELD hmap MEMVAR BLANK
CURSORJ = OBJNUM(m.hmap)
ENDIF
GO m.oldrec
ENDIF
SHOW GETS

```

_ _QKF0VR9C5 m.Save VALID
Function Origin:
From Platform: MS-DOS
From Screen: HMAP, Record Number: 5
Variable: m.Save
Called By: VALID Clause
Object Type: Push Button
Snippet Number: 3

```

```

FUNCTION _qkf0vr9c5 && m.Save VALID
#REGION 1
m.notsave = .f.
IF m.save = 1 && Selected Save Button
=errmsg("Data empty, could not save!",2)
CURSORJ = OBJNUM(m.hmap)
m.notsave = .t.
ELSE
IF m.adding && Adding a new record
APPEND BLANK
GATHER MEMVAR
ELSE
IF m.change && Changing an old record
GATHER MEMVAR
ENDIF
ENDIF
ELSE
SCATTER MEMVAR
ENDIF
SHOW GETS
IF m.notsave
SHOW GET m.hmap enabled
SHOW GET action disabled
SHOW GET SAVE enabled
ELSE
SHOW GET m.hmap disabled
SHOW GET action enabled

```



```

1  * *****
2  *
3  * Procedure file: C:\HAZMAT\GMM\WORK\MEMOWIN.SPR
4  *
5  * System: Hazardous Material Life Cycle Cost Model
6  * Author: Naval Health Research Center
7  * Copyright (c) 1993 Naval Health Research Center
8  * Last modified: 09/10/93 8:12
9  *
10 * Procs & Fncts: QIG01X0XD()
11 * : QIG01XPJO()
12 *
13 * Calls: QIG01X0XD() (function in MEMOWIN.SPR)
14 * : QIG01XPJO() (function in MEMOWIN.SPR)
15 *
16 * Documented 09/14/93 at 08:20 FoxDoc version 2.10f
17 * *****
18 *
19 *
20 * 08/17/93 MEMOWIN.SPR 08:50:04
21 *
22 *
23 *
24 * G. Pang
25 *
26 * Copyright (c) 1993 Company Name
27 * Address
28 * City, Zip
29 *
30 * Description:
31 * This program was automatically generated by GENSCRN.
32 *
33 * *****
34
35 PARAMETERS TEXT, TITLE
36 DO CASE
37 CASE _dos
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 #REGION 1
48 PRIVATE ALL
49 DO CASE
50 CLEAR
51 CASE PARAMETER() = 0
52 m.title = ""
53 m.text = ""
54 CASE PARAMETER() = 1
55 m.title = ""
56 m.text = ""
57 ENDCASE
58
59 #REGION 0
60 REGIONAL m.curarea, m.talkstat, m.compstat
61
62 IF SET("TALK") = "ON"
63 SET TALK OFF
64

```

```

65
66 m.talkstat = "ON"
67 ELSE
68 m.talkstat = "OFF"
69 ENDIF
70 m.compstat = SET("COMPATIBLE")
71 SET COMPATIBLE FOXPLUS
72
73 m.curarea = SELECT()
74
75 *****
76 *
77 * MS-DOS Window definitions
78 *
79 *****
80 *
81
82 IF NOT HEXIST("memowin") ;
83 OR UPPER(UTITLE("MEMOWIN")) == "MEMOWIN.PJX" ;
84 OR UPPER(UTITLE("MEMOWIN")) == "MEMOWIN.SCX" ;
85 OR UPPER(UTITLE("MEMOWIN")) == "MEMOWIN.MNX" ;
86 OR UPPER(UTITLE("MEMOWIN")) == "MEMOWIN.PRG" ;
87 OR UPPER(UTITLE("MEMOWIN")) == "MEMOWIN.FRX" ;
88 OR UPPER(UTITLE("MEMOWIN")) == "MEMOWIN.QPR" ;
89
90 DEFINE WINDOW memowin ;
91 FROM INT((SROW()-26)/2),INT((SCOL()-80)/2) ;
92 TO INT((SROW()-26)/2)+25,INT((SCOL()-80)/2)+79 ;
93 NOFLOAT ;
94 NOCLOSE ;
95 SHADOW ;
96 minimize ;
97 NONE ;
98 COLOR SCHEME 1
99
100 ENDIF
101
102 *****
103 *
104 * MEMOWIN/MS-DOS Screen Layout
105 *
106 *****
107
108 #REGION 1
109 IF VISIBLE("memowin")
110 ACTIVATE WINDOW memowin SAME
111 ELSE
112 ACTIVATE WINDOW memowin NOSHOW
113 ENDIF
114 @ 1,2 GET m.title ;
115 SIZE 1,74 ;
116 DEFAULT " " ;
117 PICTURE "a1" ;
118 DISABLE
119 @ 2,2 EDIT m.text ;
120 SIZE 20,77,0 ;
121 DEFAULT " " ;
122 SCROLL
123 @ 23,26 GET m.save ;
124 PICTURE "a=HT \<Print;\<Another;\<Quit" ;
125 SIZE 1,91 ;
126 DEFAULT " " ;
127 VALID _qig01x0xd()
128
129 IF NOT VISIBLE("memowin")
130 ACTIVATE WINDOW memowin
131

```



```

263 SELECT (m.curarea)
264
265 #REGION 0
266 IF m.talkstat = "ON"
267 SET TALK ON
268 ENDIF
269 IF m.compstat = "ON"
270 SET COMPATIBLE ON
271 ENDIF
272
273 *****
274 *
275 *
276 *
277 *
278 *
279 *
280 *
281 *
282 *
283 *
284 *
285 *
286 *
287 *
288 *
289 *
290 *
291 *
292 *
293 *
294 *
295 *
296 *
297 *
298 *
299 *
300 *
301 *
302 *
303 *
304 *
305 *
306 *
307 *
308 *
309 *
310 *
311 *
312 *
313 *
314 *
315 *
316 *
317 *
318 *
319 *
320 *
321 *
322 *
323 *
324 *
325 *
326 *

```

Function: _QIG0IXOXD()

Called by: MEMOIN.SPR
: MEMOIN.PRG

Calls: V_PRINT.SPR

```

312 *****
313 *
314 *
315 *
316 *
317 *
318 *
319 *
320 *
321 *
322 *
323 *
324 *
325 *
326 *

```

Function: _QIG0IXOXD() && m.save VALID

#REGION 1

IF m.save = 1 && Print

DO W_PRINT.SPR WITH m.text

RETURN m.save

ENDIF

IF m.save = 2 && Another

RETURN m.save

ENDIF

IF m.save = 3 && Quit

RETURN m.save

ENDIF

```

327 *
328 *
329 *
330 *
331 *
332 *
333 *
334 *
335 *
336 *
337 *
338 *
339 *
340 *
341 *
342 *
343 *
344 *
345 *
346 *
347 *
348 *
349 *
350 *
351 *
352 *
353 *
354 *
355 *
356 *
357 *
358 *
359 *
360 *
361 *
362 *

```

Function: _QIG0IXPJQ()

Called by: MEMOIN.SPR
: MEMOIN.PRG

Calls: V_PRINT.SPR

```

350 *****
351 *
352 *
353 *
354 *
355 *
356 *
357 *
358 *
359 *
360 *
361 *
362 *

```

Function: _QIG0IXPJQ() && m.save VALID

#REGION 1

IF m.save = 1 && Print

DO W_PRINT.SPR WITH m.text

m.quit = .f.

ENDIF

IF m.save = 2 && Another

m.quit = .f.

ENDIF

IF m.save = 3 && Quit

m.quit = .f.

ENDIF

*: EOF: MEMOIN.AC1

W_PRINT.AC1 12-1-93 11:32a

1

```

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132

```

10/27/93	U_PRINT.SPR	14:49:39
Author's Name		
Copyright (c) 1993 Company Name		
Address		
City, Zip		
Description: This program was automatically generated by GENSCRN.		

PARAMETERS TEXT

U_PRINT/MS-DOS Setup Code - SECTION 1

```

#REGION 1
IF PARAMETER() = 0
RETURN
ENDIF

```

```

#REGION 0
REGIONAL m.curraesa, m.talkstat, m.compstat

IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

MS-DOS Window definitions

```

IF NOT WEXIST("w_prn") ;
OR UPPER(UTITLE("w_prn")) == "w_prn.pjn" ;
OR UPPER(UTITLE("w_prn")) == "w_prn.scx" ;
OR UPPER(UTITLE("w_prn")) == "w_prn.mnx" ;
OR UPPER(UTITLE("w_prn")) == "w_prn.prg" ;
OR UPPER(UTITLE("w_prn")) == "w_prn.frx" ;
OR UPPER(UTITLE("w_prn")) == "w_prn.opr" ;
DEFINE WINDOW w_prn ;
FROM INT((SCOL()-13)/2), INT((SCOL()-43)/2) ;
TO INT((SCOL()-13)/2)+12, INT((SCOL()-43)/2)+42 ;
TITLE "Print Option" ;
FLOAT ;
NOCLOSE ;
SHADOW ;
nomimize ;

```

```
DOUBLE ;  
COLOR SCHEME 5  
  
[ENDIF]  
  
*  
*  
*  
*  
*  
*  
  
#REGION 1  
PRIVATE mchoice, mprintfile, mchoice = 1  
mfilename = "hnlcm.ta"  
mprintfile = mfilename  
  
*  
*  
*  
*  
*  
*  
  
[IF WVISIBLE("w_prn")  
    ACTIVATE WINDOW w_prn  
ELSE  
    ACTIVATE WINDOW w_p_pn  
ENDIF  
  
@ 0,0 TO 3,40  
@ 5,0 TO 8,40  
@ 6,1 SAY "Filename:"  
SIZE 1,9,0  
@ 6,11 GET mprintfile  
SIZE 1,29;  
DEFAULT " " ;  
VALID qkf0vs4nc()  
@ 10,9 GET mchoice ;  
PICTURE "q=HT OK;\n"  
SIZE 1,8,3 ;  
DEFAULT 1 ;  
VALID qkf0vs4ph()  
@ 1,17 GET mchoice ;  
PICTURE "q=RVN Fil"  
SIZE 1,11,0 ;  
DEFAULT 1 ;  
VALID qkf0vs4rm()  
@ 1,1 SAY "Print to:"  
SIZE 1,10,0  
  
[IF NOT WVISIBLE("w_prn")  
    ACTIVATE WINDOW w_p_pn  
ENDIF  
  
READ CYCLE  
  
RELEASE WINDOW w_prn  
  
#REGION 0  
[IF m.talkstat = "NON"  
SET TALK ON  
ENDIF  
[IF m.compstat = "NON"
```

```
PRINT/MS-DOS Setup Code - SECTION 2

file, filename, i

/_PRINT/MS-DOS Screen Layout

SAME
NOSHOW

ancel" ;

Printer" ;
```

--	--

12/01/93	HMENU.MPR	14:37:41
Author's Name Copyright (c) 1993 Company Name Address City, Zip Description: This program was automatically generated by GENMENU.		

Menu Definition

```

1  *
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 define pad _msm_sysm of _msysmenu prompt "\\System" color scheme 3 ;
31   key alt+s, ""
32 define pad _msm_file of _msysmenu prompt "\\File" color scheme 3 ;
33   key alt+f, ""
34 define pad _msm_edit of _msysmenu prompt "\\Edit" color scheme 3 ;
35   key alt=e, ""
36 define pad _msm_data of _msysmenu prompt "\\Database" color scheme 3
37   => ;
38   key alt+d, ""
39 define pad _msm_recrd of _msysmenu prompt "\\Record" color scheme 3 ;
40   key alt+r, ""
41 define pad _msm_prog of _msysmenu prompt "\\Program" color scheme 3 ;
42   key alt+p, ""
43 define pad _msm_windo of _msysmenu prompt "\\Window" color scheme 3 ;
44   key alt+w, ""
45 on pad _msm_sysm of _msysmenu activate popup _msysmenu
46 on pad _msm_file of _msysmenu activate popup _mfile
47 on pad _msm_edit of _msysmenu activate popup _mdata
48 on pad _msm_recrd of _msysmenu activate popup _mrecord
49 on pad _msm_prog of _msysmenu activate popup _mprog
50 on pad _msm_windo of _msysmenu activate popup _mwindow
51
52 define popup _msysmenu margin relative shadow color scheme 4
53 define bar _mst help of _msysmenu prompt "\\Help...";
54   key f1, "h"
55 define bar _mst_sp100 of _msysmenu prompt "\\-"
56 define bar _mst_calcu of _msysmenu prompt "\\Calculator"
57 define bar 2 of _msysmenu prompt "\\-"
58 define bar 5 of _msysmenu prompt "\\HMLCCM"
59 on bar 5 of _msysmenu activate popup hmlccm
60
61 define popup hmlccm margin relative shadow color scheme 4
62 define bar 1 of hmlccm prompt "\\Cost Analysis"
63 define bar 2 of hmlccm prompt "\\OSHA 2-Table"
64 define bar 3 of hmlccm prompt "\\Reference Material"
65 define bar 4 of hmlccm prompt "\\System \\Maintenance"

```

```

66 define bar 5 of hmlccm prompt "\\Set Parameters";
67   skip for (m,pass) != "SYSHAZ"
68 on bar 1 of hmlccm activate popup costanalys
69 on selection bar 2 of hmlccm do hmtab
70 on selection bar 3 of hmlccm do hmtabs
71 on bar 4 of hmlccm activate popup systemmain
72 on bar 5 of hmlccm activate popup setparamet
73
74 define popup costanalys margin relative shadow color scheme 4
75 define bar 1 of costanalys prompt "\\Build Hazmat Scenario"
76 on selection bar 1 of costanalys do hmsc
77
78 define popup systemmain margin relative shadow color scheme 4
79 define bar 1 of systemmain prompt "\\Back-Up (floppy)"
80 define bar 2 of systemmain prompt "\\djp-Load Data"
81 on selection bar 1 of systemmain ;
82   do qledvcgc ;
83   in Tocfile("WORK\\HMENU" , "MPX;MPR|FXP;PRG" , "Where is HMENU?")
84 on selection bar 2 of systemmain ;
85   do qledvcgq ;
86   in Tocfile("WORK\\HMENU" , "MPX;MPR|FXP;PRG" , "Where is HMENU?")
87
88 define popup setparamet margin relative shadow color scheme 4
89 define bar 1 of setparamet prompt "\\Materials";
90   key ctrl+m, "CTRL+M"
91 define bar 2 of setparamet prompt "\\Life Cycle Phase";
92   key ctrl+l, "CTRL+L"
93 define bar 3 of setparamet prompt "\\Process";
94   key ctrl+w, "CTRL+W"
95 define bar 4 of setparamet prompt "\\Exposure Type";
96   key ctrl+x, "CTRL+X"
97 define bar 5 of setparamet prompt "Cost \\Factors ";
98   key ctrl+f, "CTRL+F"
99 define bar 6 of setparamet prompt "Cost Factor \\Elements ";
100  key ctrl=e, "CTRL+E"
101 define bar 7 of setparamet prompt "Cost Factor Element \\Items";
102  key ctrl+i, "CTRL+I"
103 define bar 8 of setparamet prompt "\\Build Hazmat Table";
104  key ctrl+b, "CTRL+B"
105 on selection bar 1 of setparamet do hmat.spr
106 on selection bar 2 of setparamet do hmlc.spr
107 on selection bar 3 of setparamet do hmp.spr
108 on selection bar 4 of setparamet do hmet.spr
109 on selection bar 5 of setparamet do hmc.spr
110 on selection bar 6 of setparamet do hmcfe.spr
111 on selection bar 7 of setparamet do hmcfe1.spr
112 on selection bar 8 of setparamet do hmtab.spr
113
114 define popup _mfile margin relative shadow color scheme 4
115 define bar _mfi_setup of _mfile prompt "pr<inter Setup..."
116 define bar _mfi_print of _mfile prompt "\\Print..."
117 define bar _mfi_sp300 of _mfile prompt "\\-"
118 define bar 4 of _mfile prompt "\\Quit"
119 on selection bar 4 of _mfile do _quit in hminit
120
121 define popup _medit margin relative shadow color scheme 4
122 define bar _med_undo of _medit prompt "\\Undo";
123   key ctrl+u, "U"
124 define bar _med_redo of _medit prompt "\\Redo";
125   key ctrl+r, "R"
126 define bar _med_sp100 of _medit prompt "\\-"
127 define bar _med_cut of _medit prompt "Cu<t";
128   key ctrl+x, "X"
129 define bar _med_copy of _medit prompt "\\Copy";
130   key ctrl+c, "C"
131 define bar _med_paste of _medit prompt "\\Paste";

```

```

132 key ctrl+v, "v"
133 define bar _med_clear of _medit prompt "Clear"
134 define bar _med_sp200 of _medit prompt "\-"
135 define bar _med_acta of _medit prompt "Select \<All";
136 key ctrl+a, "A"
137 define bar _med_sp300 of _medit prompt "\-"
138 define bar _med_goto of _medit prompt "Goto \<Line..."
139 define bar _med_find of _medit prompt "\<Find...";
140 key ctrl+f, "F"
141 define bar _med_finda of _medit prompt "Find \<gain";
142 key ctrl+g, "G"
143 define bar _med_repl of _medit prompt "R\<eplace And Find Again";
144 key ctrl+e, "E"
145 define bar _med_repla of _medit prompt "Replace All"
146 define bar _med_sp400 of _medit prompt "\-"
147 define bar _med_pref of _medit prompt "Prefere\<nces..."
148
149 define popup _mdata margin relative shadow color scheme 4
150 define bar _mda_brow of _mdata prompt "\<Browse"
151 define bar _mda_sp100 of _mdata prompt "\-"
152 define bar _mda_sort of _mdata prompt "\<Sort..."
153 define bar _mda_total of _mdata prompt "\<Total..."
154 define bar _mda_sp200 of _mdata prompt "\-"
155 define bar _mda_avg of _mdata prompt "A\<verage..."
156 define bar _mda_count of _mdata prompt "C\<ount..."
157 define bar _mda_sum of _mdata prompt "S\<um..."
158 define bar _mda_calc of _mdata prompt "Calculat\<e..."
159 define bar _mda_reprt of _mdata prompt "\<Report..."
160
161 define popup _mrecord margin relative shadow color scheme 4
162 define bar _mrc_goto of _mrecord prompt "\<Goto..."
163 define bar _mrc_locat of _mrecord prompt "\<Locate..."
164 define bar _mrc_cont of _mrecord prompt "\<Continue";
165 key ctrl+k, "K"
166 define bar _mrc_seek of _mrecord prompt "\<Seek..."
167 define bar _mrc_sp200 of _mrecord prompt "\-"
168 define bar _mrc_repl of _mrecord prompt "Re\<place..."
169 define bar _mrc_delet of _mrecord prompt "\<Delete..."
170 define bar _mrc_recal of _mrecord prompt "\<Recall..."
171
172 define popup _mprog margin relative shadow color scheme 4
173 define bar _mpr_canc of _mprog prompt "\<Cancel"
174 define bar _mpr_resum of _mprog prompt "\<Resume";
175 key ctrl+m, "M"
176
177 define popup _mwindow margin relative shadow color scheme 4
178 define bar _mwi_hide of _mwindow prompt "\<Hide"
179 define bar _mwi_hides of _mwindow prompt "\<Hide All"
180 define bar _mwi_showa of _mwindow prompt "Sh\<ow All"
181 define bar _mwi_clear of _mwindow prompt "Clea\<r"
182 define bar _mwi_sp100 of _mwindow prompt "\-"
183 define bar _mwi_move of _mwindow prompt "\<Move";
184 key ctrl+i, "I"
185 define bar _mwi_size of _mwindow prompt "\<Size";
186 key ctrl+f8, "F8"
187 define bar _mwi_zoom of _mwindow prompt "\<Zoom 1";
188 key ctrl+f10, "F10"
189 define bar _mwi_min of _mwindow prompt "Z\<oom 1";
190 key ctrl+f9, "F9"
191 define bar _mwi_rotat of _mwindow prompt "\<Cycle";
192 key ctrl+f11, "F11"
193 define bar _mwi_color of _mwindow prompt "Co\<lor..."
194 define bar _mwi_sp200 of _mwindow prompt "\-"
195 define bar _mwi_cmd of _mwindow prompt "Command";
196 key ctrl+f2, "F2"
197 define bar _mwi_debug of _mwindow prompt "\<Debug"

```

```

198 define bar _mwi_trace of _mwindow prompt "\<Trace"
199 define bar _mwi_view of _mwindow prompt "\<View"
200
201 *
202 *
203 *
204 *
205 *
206 *
207 *
208
209 droppedead = .f.
210
211 *
212 *
213 *
214 *
215 *
216 *
217 *
218 *
219 *
220 *
221 *
222 *
223 *
224 *
225
226 procedure ql0vcqdc
227 * Do Back-Up on floppy
228 backup()
229 return
230 * EOF Back-Up
231
232 *
233 *
234 *
235 *
236 *
237 *
238 *
239 *
240 *
241 *
242 *
243 *
244 *
245 *
246
247 procedure ql0vcqdcg
248 * BACK TO VAX FROM REMOTE SITE
249 *errmsg("This option is not available in this version.")
250 *vax_back()
251 return
252
253 *: EOF: HMENU.AC2

```

Cleanup Code & Procedures

```

_QLEOVCDQC ON SELECTION BAR 1 OF POPUP systemmain
Procedure Origin:
From Menu: HMENU.MPR, Record: 18
Called By: ON SELECTION BAR 1 OF POPUP systemmain
Prompt: Back-Up (floppy)
Snippet: 1

```

```

_QLEOVCDQCG ON SELECTION BAR 2 OF POPUP systemmain
Procedure Origin:
From Menu: HMENU.MPR, Record: 19
Called By: ON SELECTION BAR 2 OF POPUP systemmain
Prompt: Up-Load Data
Snippet: 2

```

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE OCT 1993		3. REPORT TYPE AND DATE COVERED FINAL OCT 92 - SEPT 93
4. TITLE AND SUBTITLE HAZARDOUS MATERIAL LIFE-CYCLE COST MODEL TECHNICAL MANUAL, VERSION 1.2			5. FUNDING NUMBERS Program Element: REIMB Work Unit Number: NAVFAC.WR.1082W	
6. AUTHOR(S) H. LY AND G. PANG				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Health Research Center P. O. Box 85122 San Diego, CA 92186-5122			8. PERFORMING ORGANIZATION Technical Document No. 93-3C	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Medical Research and Development Command National Naval Medical Center Building 1, Tower 2 Bethesda, MD 20889-5044			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This technical manual contains the information on the program code, data elements, and file structures needed to maintain the Hazardous Material Life-Cycle Cost Model. This documentation was created using the FoxDoc Version 2.5a program.				
14. SUBJECT TERMS LIFE-CYCLE COST MODEL COST-BENEFIT ANALYSIS TECHNICAL HAZARDOUS MATERIALS HAZARDOUS WASTE			15. NUMBER OF PAGES 158	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited	